VISUAL EVENT DESCRIPTION IN VIDEOS

Mehrsan Javan Roshtkhari

Department of Electrical and Computer Engineering
McGill University, Montréal

December 2014

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

To my parents, Bahar, and anyone who taught me to believe in myself. Words cannot express how grateful I am to them.

Abstract

This thesis focuses on monitoring non-specific and unconstrained activities and events in videos in order to build a complete scene understanding system. The particular emphasis in this work is based on the spatio-temporal context of the scene. This thesis proposes a unique solution using a hierarchical framework of video fragments to create a dynamically changing model of the scene. The model is then used to simultaneously detect and localize an event of interest, detect abnormal (rare) events, and track all moving objects in the scene. The approach can be considered as an extension to the original Bag-of-Video-Words approaches in which a spatio-temporal scene configuration comes into play. It does not require prior knowledge about actions and events, background subtraction, motion estimation or tracking. It is also robust to spatial and temporal scale changes, as well as some deformations. The hierarchical algorithm uses a probabilistic framework to code a video as a compact set of local spatio-temporal visual features, while considering their spatio-temporal compositions in order to account for the scene context. A significant aspect of the methodology is the way that we represent scene information while keeping the computational cost low enough for real-time implementation using the current hardware resources.

Given the adaptive shape- and motion-based model, the events can be described and localized in the videos. These events are interpreted by a complete scene understanding system that uses different inference mechanisms and learning strategies to describe ongoing events in a video, identify abnormal patterns is space and time, find similar videos to a query based on their contents, and track all moving objects in the

scene without using any object detection method. We have extensively tested all our system on popular benchmarks and shown that they are both effective and robust for all of the aforementioned tasks. Moreover, the results are highly competitive with state-of-the-art methods. However, a major advantage of our approach is that it does not require any feature analysis, background/foreground segmentation and tracking, and is susceptible to online real-time analysis.

Abrégé

Cette thèse porte sur le suivi des activités et des événements non spécifiques et sans contraintes dans les vidéos afin de construire un système de compréhension complète de scène. Ce travail porte une importance particulière sur le contexte spatio-temporel de la scène. Cette thèse propose une solution unique qui utilise un cadre hiérarchique de fragments de vidéo pour créer un modèle de changement dynamique de la scène. Le modèle est ensuite utilisé pour détecter et localiser simultanément un événement d'intérêt, détecter les événements anormaux (rares) et suivre tous les objets en mouvement dans la scène. L'approche peut être considérée comme une extension de l'approche originale Bag-of-Video-Words dans laquelle une configuration de la scène spatio-temporelle entre en jeu. Elle ne nécessite pas de connaissance préalable sur les actions et les événements, la soustraction d'image de fond, l'estimation de mouvement ou de suivi. Elle est également robuste aux changements d'échelle spatiale et temporelle ainsi que certaines déformations. L'algorithme hiérarchique utilise un cadre probabiliste pour coder une vidéo comme un ensemble compact de caractéristiques visuelles spatio-temporelles locales, tout en tenant compte de leurs compositions spatio-temporelles afin de tenir compte du contexte de la scène. Un aspect important de la méthodologie est la façon dont l'information de la scène est représentée tout en gardant un niveau minimal de calcul mais suffisant pour la mise en œuvre en temps réel en utilisant les ressources matérielles actuelles.

Compte tenu du modèle adaptatif basé sur la forme et le mouvement, les événements peuvent être décrits et localisés dans les vidéos. Ces événements sont interprétés

par un système de compréhension complète de scène qui utilise différents mécanismes d'inférence et de stratégies d'apprentissage pour décrire les événements en cours dans une vidéo, identifie les tendances anormales dans l'espace et le temps, trouve des vidéos similaires à une requête basée sur leur contenu et fait le suivi de tous les objets en mouvement dans la scène sans utiliser de procédé de détection d'objet. Nous avons testé à plusieurs reprises notre système en entier sur des références populaires et montré qu'il est à la fois efficace et robuste pour toutes les tâches mentionnées ci-dessus. De plus, les résultats sont très compétitifs avec les méthodes à la fine pointe de la technologie. Cependant, un avantage majeur de notre approche est qu'elle ne nécessite pas d'analyse de fonction, de segmentation et de suivi d'image de fond ou de premier plan et offre l'opportunité à l'analyse en ligne.

Acknowledgements

I owe a deep debt of gratitude to everyone who supported me in the past years. Firs, I would like to thank my PhD thesis supervisor, Prof. Martin D. Levine, for giving me a home in his lab and support over the years. I am grateful for his guidance not only in research, but also in dealing with real life challenges. He encouraged me to start my own start-up company for visual event analysis, SPORTLOGiQ, based on my PhD study. I would also like to thank my thesis committee members, Prof. James J. Clark and Prof. Frank P. Ferrie, who were more than generous with their expertise and precious time. I am fortunate to have such a group of intelligent scientists, who are also great mentors.

I also owe much to my friends and colleagues at McGill university and elsewhere. My sincerest thanks to my sister, Soroor, for supporting me from the first moment that I came to Montreal. I would also like to thank all the wonderful people whom I have met during my study at McGill university. Especially, I would like to thank my magic maker fellows in SPORTLOGiQ, who help me in building close-to-impossible things. I would also like to thank Dr. A.G. Kashkooli, soon to become a doctor Arash Mohtat, and my amazing office-mates in MC-430.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Given the tremendous number of video data produced every day, there is a great demand for automated systems that analyze and understand the events in these videos. In particular, retrieving and identifying human activities in videos has become more interesting due to its potential applications in real life. These include the following practical applications: automated video surveillance systems, human-computer interaction, assisted living environments and nursing care institutions, sports interpretation, video annotation and indexing, and video summarization. This thesis focuses on monitoring non-specific and unconstrained activities in videos. We propose a unique solution for visual event understanding using a hierarchical framework of video fragments to describe objects and their motions. These are employed to simultaneously detect and localize both dominant (activities that occur on a regular basis) and rare events (activities which are not observed regularly). Then, the framework is extended to do video-to-video matching and eventually, a model free tracker is constructed to track multiple moving objects in the scene.

More specifically, the overall objectives of this research are:

(i) To detect and localize abnormal (rare) events in videos [90].

(ii) To detect and localize normal activities in video clips [91]. In this thesis, the actual label assignment and recognition of these normal events is left for further research.

(iii) To measure the similarity between two videos and video-to-video matching [88, 89].

(iv) To track all moving objects in the scene [44].

Those objectives require the ability to measure, online and adaptively, the self-similarity of a video clip or the similarity between two video clips. Our solution is based on the bag of space-time features approach in which a prescribed set of spatio-temporal video volumes is used for measuring similarity. We refer to this as the *scene context*. Figure 1.1 represents a block-diagram of the thesis structure and objectives.

The approach presented here for modeling the scene context can be considered as an extension of the original Bag-of-Video-Words approaches in which a spatio-temporal scene configuration comes into play. It imposes spatial and temporal constraints on the video fragments so that an inference mechanism can estimate the probability density functions of their arrangements. A significant aspect of the methodology is the way that we represent scene information while keeping the computational cost low enough for real-time implementation using currently available hardware resources. Moreover, it does not require lengthy training periods, object segmentation, tracking and background subtraction, with their attendant weaknesses, which form the basis for previously reported approaches. By observing a scene in real-time, the system builds a dynamically changing model of the environment. This adaptive appearance-based model, which is probabilistic in nature, is employed to describe the ongoing events. Our approach provides probabilistic graphical structures of all moving objects while simultaneously coding the spatio-temporal context of the scene in the surrounding regions. The probabilistic graphical structures are then used to find and localize different events in the scene. Therefore, a video is represented by set of events, localized in space and time, and coded by probabilistic graphical structures. Such a framework can be considered as the building block for the computer

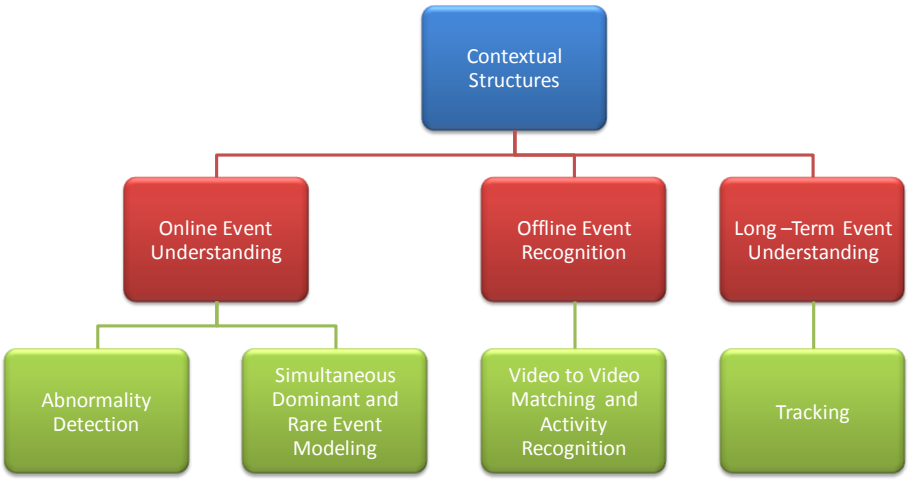


FIGURE 1.1. Thesis structure. The main focus of this thesis is to build a system for visual event description in videos. First, we introduce a framework for capturing contextual structures in videos. It uses local and global motion patterns to construct an adaptive structure of all objects in the scene. Then, this framework is employed to solve three problems: (1) Online abnormal event detection in videos and (2) Offline activity recognition and video-to-video matching, (3) Tracking multiple objects (such as persons) in the video without doing any object detection.

vision applications described earlier. For example, based on the produced probabilistic models for all events and objects in the scene, further analysis of their behaviors and interactions can be performed to produce video semantics and a complete scene description.

1.2 Event Understanding: Problem Statements And Contributions

The main problem that this thesis attempts to solve is to build a complete framework for event understanding in videos. We start from the low level image features, which are the local appearance and motion patterns. Given the thesis structure illustrated in Figure 1.1, at first a hierarchical structure is introduced to capture the scene context. Therefore, the main contribution of this thesis is to provide a unique structure to model contextual information in the conventional BOW paradigm. It uses a probabilistic framework to capture spatio-temporal configurations of video volumes.

This is achieved by estimating probability density functions of the *arrangements* of video volumes. This is explained in Chapter 2.

Then, three frameworks are built to address three of the most challenging problems in computer vision: *I*- Normal and abnormal event detection, *II*- Activity recognition and video to video matching, and *III*- Multi-object tracking using local motion patterns.

1.2.1 Terminology. Before continuing, we summarize our terminology because some of the terms are used in multiple ways in the related literature. Spatio-temporal video words refer to 3D (in XYT space) pixel level features extracted at each pixel in a video. An ensemble of video volumes refers to a large spatio-temporal region consisting of many video volumes. Low-level behaviors refer to those activities that can be localized in space and time. In this thesis, the term "event" is deemed to be more general than "activity" as it is not restricted to just humans (animate objects). To date, in the computer vision community, the term "activity" has largely been taken to be a human action performed by a single person, lasting for just a few video frames, taking up to a few seconds, and containing one or more events. Finally, by using the term "context" or "contextual information", we are referring to the relative spatio-temporal location in 3D (in XYT space) obtained by sampling video observations. In this thesis, the context of a 3D observation is taken to be a larger 3D video volume surrounding it.

1.2.2 Anomaly Detection. In recent years video surveillance systems have become very popular due to heightened security concerns and low-hardware costs. At present they are widely used in applications such as law enforcement, building security, and traffic analysis. Moreover, in most circumstances, it is necessary for humans to analyze the videos, which is inefficient in terms of effectiveness, accuracy and cost [37, 25]. In light of this, together with the tremendous number of such videos produced on a daily basis, there is a great need for a real-time automated system that detects and locates suspicious behaviors and alerts security agents.

Consequently, detecting *unusual* or *suspicious activities*, *uncommon behaviors*, or *irregular events* in a scene is the primary objective of an automated video surveillance system. We refer to this activity as *anomaly detection* because the sought-after situations are not observed frequently. Although the term *anomaly* cannot be defined explicitly, all such systems are based on the implicit assumption that events that occur occasionally are potentially suspicious, and thus may be considered as being anomalous [2, 14, 56, 65, 67, 129, 132, 11, 115, 79, 6]. Therefore, the working definition of this term in this paper is taken to be the spatio-temporal compositions in a video or set of videos with low probability of occurrence with respect to the previous observations. This implies that the anomalies are spatial, temporal, or spatio-temporal outliers that are different from the regularly observed patterns. We define the anomalies with respect to a pixel's context, meaning that a particular activity in a particular context would be an anomaly, while in another context it might be normal [115].

Thus the question arises as to how a set of *new* observations can be classified as being either normal or abnormal? Perhaps this is the most difficult challenge in this research. Among the proposed solutions in the literature, we believe that the most promising and reliable answer to this question should simultaneously determine both normal and abnormal compositions. Clearly, the only difference between these is that the likelihood of occurrence of the latter will be much smaller than that of the former. In light of this definition, it is possible to accomplish this task by considering the problem as one of reconstruction, as was done in [14]. Consequently, possessing a few video samples of a normal event ("training set"), a new normal observation would have high likelihood, while an abnormal event would have low likelihood. In either case, video compositions should be capable of being *reconstructed* by finding similar regions to those already found in these videos. Here we deal only with the abnormal events. We present a *fast* online unsupervised method for anomaly detection in videos, based on spatio-temporal video volume reconstruction, while using both local and global compositional information regarding the volumes. The main characteristics of the proposed framework for abnormality detection are as follows:

- Given the contextual model of the scene, anomalies are defined as those spatio-temporal compositions in a video or set of videos having very low probability of occurrence.
- It significantly reduces the size of the database for finding *similar* examples to a new observation while retaining summary information, thereby speeding up the process and making it real-time.
- It uses an online unsupervised incremental method in order to update the probability distribution functions of the normal events. Thus, our method can adaptively learn newly observed normal patterns.

1.2.3 Simultaneous Dominant And Rare Event Detection. Normal events observed in a scene will be referred to as the "dominant" behaviors. These are events that have a higher probability of occurrence than others in the video and hence generally do not attract much attention. We can further categorize dominant behaviors into two classes. In the literature on human attention processes, the first usually deals with foreground activities in [9, 8, 30, 49] while the second describes the scene background[1]. Typically, the detection of the latter is more restrictively referred to as background subtraction, which is the building block of almost all computer vision algorithms. However, dominant behavior detection is more general and more complicated than background subtraction, since it includes the scene background while not being limited to it. Thus the manner in which these two human attention processes differ is the way that they use the scene information. Most background subtraction methods are based on the principle that the photometric properties of the scene in the video, such as luminance and color, are stationary. In contrast, dominant behavior understanding can be seen as a generalization of the classical background subtraction method in which all of the dynamic contents of the video come into play as well.

[1]By definition, the background consists of pixels in the video frames whose photometric properties, such as luminance and color, are either static or stationary with respect to time.

In order to learn both normal and abnormal patterns, a new framework is introduced in this thesis. The main characteristics of our approach are as follows:

- The spatio-temporal contextual information in a scene is decomposed into separate spatial and temporal *contexts*, which make the algorithm capable of detecting purely spatial or temporal activities, as well as spatio-temporal abnormalities.
- High level activity modeling and low level pixel change detection are performed simultaneously by a single algorithm. Thus the computational cost is reduced since the need for a separate background subtraction algorithm is eliminated. This makes the algorithm capable of understanding behaviors of different complexity.
- The algorithm adaptively learns the behavior patterns in the scene in an online manner. As such, the approach is a preferable choice for visual surveillance systems.
- A major benefit of the algorithm is its *extendibility*, which is achieved by hierarchical clustering. This makes the algorithm capable of understanding dominant behaviors of different complexity.

1.2.4 Activity Recognition. Human activity analysis is required for video surveillance systems, human-computer interaction, sports interpretation, and video retrieval for content-based search engines [80, 107]. Moreover, given the tremendous number of video data available online these days, there is a great demand for automated systems that analyze and understand the contents of these videos. Recognizing and localizing human actions in a video is a primary component of such a system, and also the most important, as it significantly affects its performance. Although many methods exist to determine human actions in highly controlled environments, this task remains a challenge in real world environments due to camera motion, cluttered backgrounds, occlusion, and scale/ viewpoint/ perspective variations [74, 95, 113, 114].

Moreover, the same action performed by two persons can appear to be very different. In addition, clothing, illumination and background changes can increase this dissimilarity [14, 69, 99].

One of the goals of this thesis is to address the problem of *action recognition* and *localization* in real environments using a hierarchical probabilistic video-to-video matching framework. This problem is also referred to as *action spotting* [26]. To achieve this, we have developed a fast data-driven approach, which finds similar videos in a "target" set to a single labelled "query" video. Assuming that the latter contains an action of interest, e.g., walking, we find all videos in the target set that that are similar to the query, which implies the same activity. This video-to-video comparison also makes it possible to *label* activities, the so-called action classification problem. The major benefit of our approach is that it does not require long video training sequences, object segmentation, tracking or background subtraction. The method can be considered as an extension to the original *Bag of Video Words* (BOW) approach for action recognition. The main characteristics of this algorithm and the contributions are as follows:

- We introduce a hierarchical codebook structure for action detection and labelling. This is achieved by considering a large volume containing many *Spatio-Temporal Video Volumes* (STVs) and constructing a probabilistic model of this volume to capture the spatio-temporal configurations of STVs. Subsequently, similarity between two videos is calculated by measuring the similarity between spatio-temporal video volumes and their compositional structures.
- We select the salient pixels in the video frames by analyzing codewords obtained at the highest level of the hierarchical codebook's structure. This depends on both the local spatio-temporal video volumes and their compositional structures. This approach differs from conventional background subtraction and salient point detection methods.

1.2.5 Tracking. Visual tracking is an important task within the field of computer vision. The proliferation of high-end computers, the availability of high quality video cameras, and the increasing need for automated video analysis have generated a great deal of interest in visual tracking algorithms. The state of this art has advanced significantly in the past 30 years [121, 119]. Visual tracking is a process of continuously inferring the state of a target in a video sequence, which is modeled within the framework of detection and data association. Usually, it is formulated as a search problem that aims at finding the candidate most similar to a target. Tracking is a relatively solved problem when the objects in a scene are isolated from each other and easily distinguishable from the background. However, in complex and crowded scenes of people there are many objects with similar appearance that can occlude each other. In addition, occlusions can also be the result of static objects in a scene. Therefore, multiple object tracking remains a challenging problem in computer vision [22]. Decades of research on this topic have produced a diverse set of approaches and a rich collection of tracking algorithms. Readers can refer to [121] and [119] for a review of the state-of-the-art in object tracking and a detailed analysis and comparison of various representative methods.

In the majority of the traditional approaches, only the object itself and/or its background are modeled. Hence, significant progress has been made in tracking specifically known objects. For example, many research articles have addressed face, human body, head, and rigid object tracking, which can be categorized within this paradigm of detect-then-track. This is usually done by constructing a tracker based on a pre-trained detection and recognition mechanism for the objects of interest and is based on appearance modeling of the target [45, 64, 102]. However, suppose that there is no prior knowledge about the object to be tracked. Its detection cannot be performed. These tracking methods are referred to as "generic object tracking" or "model-free tracking". Since manually annotating sufficient numbers of examples of all objects in the world is prohibitively expensive and impractical, recently, approaches for model-free tracking have received increased interest [57, 63]. Model-free tracking

is a challenging task because there is little information available about the object to be tracked [63]. Another challenge in multi-target model-free tracking is the presence of an unknown and ever changing number of targets.

Here we concentrate on creating long-term trajectories for unknown moving objects by using a model-free tracking algorithm. As opposed to existing "tracking by detection" algorithms [47, 118], no object detection is involved in our method. Therefore appearance plays no role. Instead, each individual object is tracked by modeling the temporal relationship between sequentially occurring local motion patterns. This is achieved by constructing two separate sets of initial tracks that code local and global motion patterns in videos. The local motion patterns are obtained by analyzing spatially and temporally varying structures in videos. Our proposed approach is capable of learning long-term trajectories of any moving object in a video without using any prior knowledge about the objects (object detection). This is accomplished by creating local trajectories of regions that have similar motion patterns, while also considering their neighboring regions (contextual information). Therefore, this algorithm is a complete *bottom up* tracking method that only employs a hierarchical codebook structure of local motion patterns as the *observations*.

1.3 How To Read This Dissertation

This thesis is structured in the same way as illustrated in Figure 1.1. Therefore, Chapter 2 describes the fundamental aspects of how to code contextual information in video. This forms the building block of the other solutions for content-based visual analysis described in this thesis. Each chapter of this thesis focuses on one aspect of visual event understanding and therefore, it can be read independently from the rest of the thesis. More specifically, a reader who is interested in automated abnormality detection in videos can read Chapters 2, 3 and 4. Similarly, Chapters 2 and 5 are the only chapters that are necessary to read for event recognition in videos. Finally, Chapters 2 and 6 propose a general framework for multi-object tracking based on low level local motion patterns.

In Chapter 7 we conclude this dissertation by providing a through discussion on what can be achieved by constructing contextual graphs of local motion patterns for video analysis. The strengths and weaknesses of the current framework are described and future research ideas are proposed.

Chapter 2

Visual Event Detection: Context Is Important

2.1 Introduction

Event understanding in videos is the key element of all computer vision systems either in the context of visual surveillance or action recognition. Therefore, an event or activity must be represented in such a way that it retains all of the important visual information in a compact structure. In the context of human behavior analysis, many studies have focused on the action recognition problem by invoking human body models, tracking-based methods, and local descriptors [80]. The early work often depended on tracking [82, 83, 125, 110], in which humans, body parts, or some interest points were tracked between consecutive frames to obtain the overall appearance and motion trajectory. Clearly, the performance of these algorithms is highly dependent on tracking, which sometimes fails for real world video data [119].

Alternatively, shape template matching has been employed for activity recognition; e.g., 2D shape matching [122] or its 3D extensions, optical flow matching [36, 101, 29]. In this case, action templates are constructed to model the actions and used to locate similar motion patterns. Other studies have combined both shape and motion features to achieve more robust results [50, 46], claiming that this representation is somewhat robust to object appearance [50]. In a more recent study, [46],

shape and motion descriptors are employed to construct a shape-motion prototype for human activities in a hierarchical tree structure and action recognition is performed in the joint shape and motion feature space. Although it seems that the previous approaches are likely well suited to action localization, they do require a priori high-level representations of the human motion. Moreover, they depend on such image pre-processing stages as segmentation, object tracking, and background subtraction [120], which are extremely challenging in real-world unconstrained environments.

In the context of abnormality detection, approaches that focus on local spatio-temporal abnormal patterns are very popular. These rely mainly on extracting and analyzing local low-level visual features, such as motion and texture, either by constructing a pixel-level background model and behavior template [53, 49, 9, 30] or by employing spatio-temporal video volumes, STVs, (dense sampling or interest point selection) [14, 21, 27, 34, 50, 54, 74, 75, 84, 95, 99, 11, 52, 91]. In large part, the former relies on an analysis of the activity pattern (busy-idle rates) of each pixel in each frame as a function of time. These are employed to construct a background model, either by analyzing simple color features at each pixel [53] or more complex motion descriptors [49, 30]. More advanced approaches also incorporate the spatio-temporal compositions of the motion-informative regions to build background and behavior templates [9, 70, 91] that are subtracted from newly observed behaviors in order to detect an anomaly. In [129], dynamic behaviors are modeled using spatio-temporal oriented energy filters to construct an activity pattern for each pixel in a video frame. Generally, the main drawback associated with these methods is their locality. Since the activity pattern of a pixel cannot be used for behavioral understanding, their applicability in surveillance systems is restricted to the detection of local temporal phenomena [129, 49].

2.2 BOW: An Interesting But Incomplete Representation

In a completely different vein, models based on a bag of local visual features have recently been studied extensively and shown promising results for action recognition

[14, 16, 21, 50, 54, 74, 75, 95, 99, 120, 123]. The idea behind the Bag of Visual Words (BOW) comes from text understanding problems. The understanding of a text document relies on the interpretation of its words. Therefore, high-level document understanding requires low-level word interpretation. Analogously, computers can accomplish the task of visual recognition in a similar way.

In general, visual event understanding approaches based on BOW extract and quantize the video data to produce a set of video volumes that form a "visual vocabulary". These are then employed to form a *visual dictionary*. We refer to this visual dictionary as a "codebook". Using the codebook, visual information is converted into an intermediate representation, upon which sophisticated models can be designed for recognition. Codebooks are constructed by applying "coding" rules to the extracted visual vocabularies. The coding rules are essentially clustering algorithms which form a group of visual words based on their similarity [91]. Each video sequence is then represented as a histogram of codeword occurrences and the obtained representation is fed to an inference mechanism, usually a classifier.

A major advantage of using volumetric representations of videos is that it permits the localization and classification of actions using data-driven nonparametric approaches instead of requiring the training of sophisticated parametric models. In the literature, action inference is usually determined by using a wide range of classification approaches, ranging from sub-volume matching [101], nearest neighbor classifiers [15] and their extensions [126], support [21] and relevance vector machines [75]. Driven by the success of latent topic models on text understanding, researchers have also applied topic models to the task of visual scene understanding and action recognition. Using the BOW paradigm, the obtained histograms of visual word occurrence are employed to form a generative probabilistic model of the activities. For instance, Bissacco et al. apply the Latent Dirichlet Allocation (LDA) to detect humans and estimate poses from single images in [13]. In addition, some more complicated inference mechanism have been developed by employing probabilistic Latent Semantic Analysis (pLSA) [74] for human action categorization. However, Boiman et al. [15]

have shown that a rather simple nearest-neighbor image classifier in the space of the local image descriptors is as efficient as these more sophisticated classifiers. This also implies that the particular classification method chosen is not as important as might be thought, and that the main challenge for action representation is using appropriate features.

However, we note that classical BOW approaches suffer from a significant challenge. That is, the video volumes are grouped (clustered) solely based on their similarity, in order to reduce the vocabulary size. Unfortunately, this destroys the compositional information concerning the relationships between volumes [58, 74]. In addition, although the generative probabilistic frameworks such as PLSA and LDA can discover different topics corresponding to different actions, they fail to take into consideration the contextual relationship between features. Thus, the likelihood of each video volume is calculated as its similarity to the other volumes in the dataset, without considering the spatio-temporal properties of the neighboring contextual volumes. This makes the classical BOW approach[1] excessively dependent on very local data and unable to capture significant spatio-temporal relationships. In addition, it has been shown recently that detecting actions using an "order-less" BOW does not produce acceptable recognition results [14, 16, 54, 56, 58, 62, 130]. We discuss this issue in section 2.3 in more detail.

2.3 Contextual Information Matters: Hierarchical Structure

What makes the BOW approaches interesting is that they code the video as a compact set of local visual features and do not require object segmentation, tracking or background subtraction. Although an initial spatio-temporal volumetric representation of human activity might eliminate these pre-processing steps, it suffers from a major drawback: It ignores the contextual information. In other words, different

[1] Essentially the probabilistic topic models, such as the Latent Dirichlet Allocation (LDA), can also be considered as BOW approaches since they ignore the spatio-temporal order of the local features.

activities can be represented by the same visual vocabularies, even though they are completely different [14].

To overcome this challenge, contextual information must be included in the original BOW framework. One solution is to employ visual phrases instead of visual words. This has been proposed in [130] where a visual phrase is defined as a set of spatio-temporal video volumes with a specific pre-ordained spatial and temporal structure. The main drawback of this approach is that it cannot localize different activities in a video frame. Alternatively, the solution presented by Boiman and Irani [14] is to densely sample the video and store *all* video volumes for a video frame, along with their relative locations in *space* and *time*. Consequently, the likelihood of a query in an arbitrary space-time contextual volume can be computed and thereby used to determine an accurate label for an action using just simple nearest neighbor classifiers [15]. However, the main problem with this approach is that it requires excessive computational time and a considerable amount of memory to store all of the volumes as well as their spatio-temporal relationships. We present an alternative to this in the next two chapters which updates the learned structures in an online manner to adapt to the new observations and scene changes.

In addition to [14], several other methods have been proposed to incorporate spatio-temporal structure in the context of BOW. These are often based on co-occurrence matrices that are employed to describe contextual information. For example, the well-known correlogram exploits spatio-temporal co-occurrence patterns [95]. However, only the relationship between the two nearest volumes was considered. This makes the approach too local and unable to capture complex relationships between different volumes. Another approach is to use a coarse grid and construct a histogram to subdivide the space-time volumes [38]. Similarly, in [112], contextual information is added to the BOW by employing a coarse grid at different spatio-temporal scales. An alternative that does incorporate contextual information within a BOW framework is presented in [62], in which three-dimensional spatio-temporal pyramid matching is employed. While not actually comparing the compositional graphs of image

fragments, this technique is based on the original two-dimensional spatial pyramid matching of multi-resolution histograms of patch features [58]. Likewise in [92], temporal relationships between clustered patches are modeled using ordinal criteria (e.g., equals, before, overlaps, during, after, etc.) and expressed by a set of histograms for all patches in the whole video sequence. Similar to [92], in [124] ordinal criteria are employed to model spatio-temporal compositions of clustered patches in the whole video frame during very short temporal intervals. The main problem associated with this is the large size of the spatio-temporal relationship histograms and the many parameters associated with the spatio-temporal ordinal criteria. In Chapters 3, 4 and 5 the current state of the art for incorporating spatio-temporal contextual information for both abnormality detection and activity recognition will be discussed in more detail.

In this thesis, we present an alternative probabilistic framework for quantifying the arrangement of the spatio-temporal volumes at a pixel in the video. Our solution for modeling contextual information in the BOW is a hierarchical probabilistic codebook structure. This method can be considered as an extension to the original *Bag of Video Words* (BOW) approach for visual event modeling.

Given the problems of abnormality detection and activity recognition, our aim is to measure either the self similarity of a video or the similarity between two videos, the query and the target videos. Our work is based on the bag of space-time features approach in that a set of STVs is used for measuring similarity. The task consists of two main steps: visual scene representation (see Figure 2.1) and using an inference mechanism for similarity measurement. In this section, we focus on the former and Chapters 3, 4 and 5 describe the inference mechanisms for abnormality detection and activity recognition, respectively. We first explain the sampling strategy and then describe the hierarchical codebook structure.

2.3.1 Low-Level Scene Representation: BOW. The first stage of the algorithm is to represent a query video by meaningful spatio-temporal descriptors. This is achieved by dense sampling, thereby producing a large number of spatio-temporal

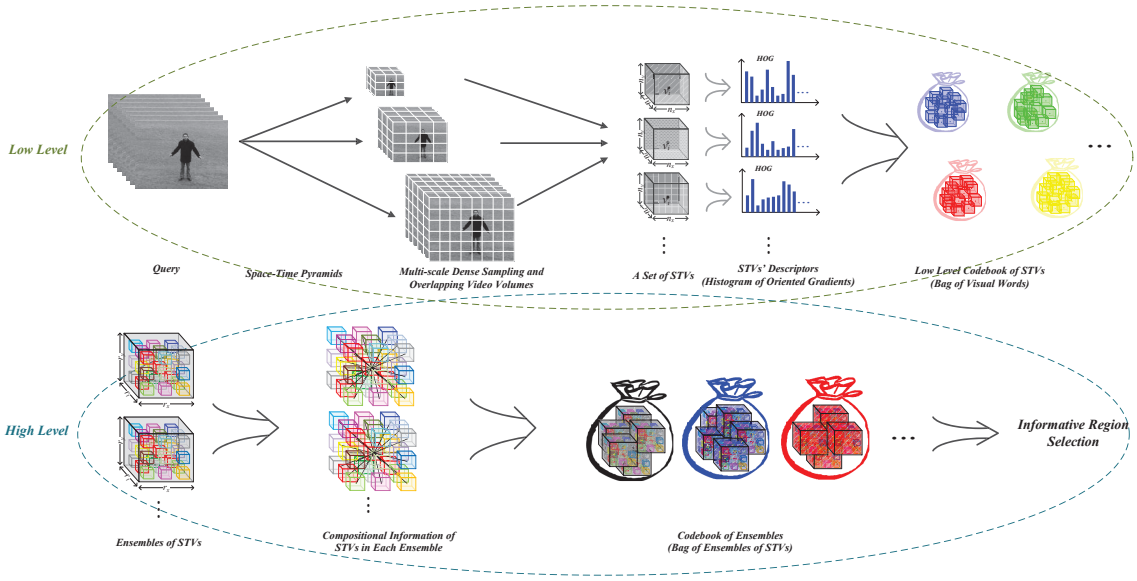


FIGURE 2.1. Overview of the scene representation and hierarchical codebook structure. First, the query video is densely sampled at different spatio-temporal scales followed by the construction of a set of overlapping spatio-temporal video volumes. Subsequently, a two level hierarchical probabilistic codebook is created for the video volumes. At the lower level of the hierarchy, similar video volumes are grouped to form a conventional low level codebook, $\mathbf{C}^{\mathcal{L}}$, but while considering the uncertainty in codeword assignment. At the higher level, a much larger spatio-temporal 3D volume around each pixel, containing many STVs, is considered in order to capture the spatio-temporal arrangement of the volumes. We refer to this graph as an ensemble of volumes. Using these graphs, similar ensembles are grouped based on the similarity between arrangements of their video volumes and yet another codebook is formed.

video volumes. Then similar video volumes are clustered to form a codebook. Since this is actually done on-line, frame-by-frame, the codebook is adaptive. The constructed codebook at this level is called the low-level codebook, as illustrated in Figure 2.1.

2.3.1.1 *Multi-Scale Dense Sampling.* Similar to all BOW approaches, 3D STVs in a video are constructed at the lowest level of the hierarchy. Although there are many methods for sampling the video for volume construction, dense sampling has been shown to be superior to the others in terms of retaining the informative features of a video [84]. Therefore, performance almost always increases with the number

of sampled spatio-temporal volumes, making dense sampling the preferable choice [111, 14].

The 3D spatio-temporal video volumes, $v_i \in \mathbb{R}^{n_x \times n_y \times n_t}$ are constructed by assuming a volume of size $n_x \times n_y \times n_t$ around each pixel (in which $n_x \times n_y$ is the size of the spatial (image) window and n_t is the depth of the video volume in time). Spatio-temporal volume construction is performed at several spatial and temporal scales of a Gaussian space-time video pyramid. This yields a large number of volumes at each pixel in the video. Figure 2.1 illustrates the process of spatio-temporal volume construction. These volumes are then characterized by a descriptor, which is the histogram of the spatio-temporal oriented gradients in the video, expressed in polar coordinates [11, 97, 91]. Assume that $G_x(x,y,t)$ and $G_y(x,y,t)$ are spatial gradients and $G_t(x,y,t)$ is the temporal gradient for each pixel at (x,y,t). The spatial gradient used to calculate the 3D gradient magnitude is normalized to reduce the effect of local texture and contrast. Hence, let:

$$\begin{aligned} G_s(x,y,t) &= \sqrt{G_x(x,y,t)^2 + G_y(x,y,t)^2}\ ,\ (x,y,t) \in v_i \\ \tilde{G}_s(x,y,t) &= \frac{G_s(x,y,t)}{\sum\limits_{(x,y,t)\in v_i} G_s(x,y,t) + \epsilon_{\max}} \end{aligned} \tag{2.1}$$

where $\tilde{G}_s$ is the normalized spatial gradient and $\epsilon_{\max}$ is a constant, set to 1% of the maximum spatial gradient magnitude in order to avoid numerical instabilities. Hence, the 3D normalized gradient is represented in polar coordinates ($M(x,y,t)$, $\theta(x,y,t)$,$\phi(x,y,t)$):

$$\begin{aligned} M(x,y,t) &= \sqrt{\tilde{G}_s(x,y,t)^2 + G_t(x,y,t)^2} \\ \theta(x,y,t) &= \tan^{-1}\left(\frac{G_y(x,y,t)}{G_x(x,y,t)}\right) \\ \phi(x,y,t) &= \tan^{-1}\left(\frac{G_t(x,y,t)}{\tilde{G}_s(x,y,t)}\right) \end{aligned} \tag{2.2}$$

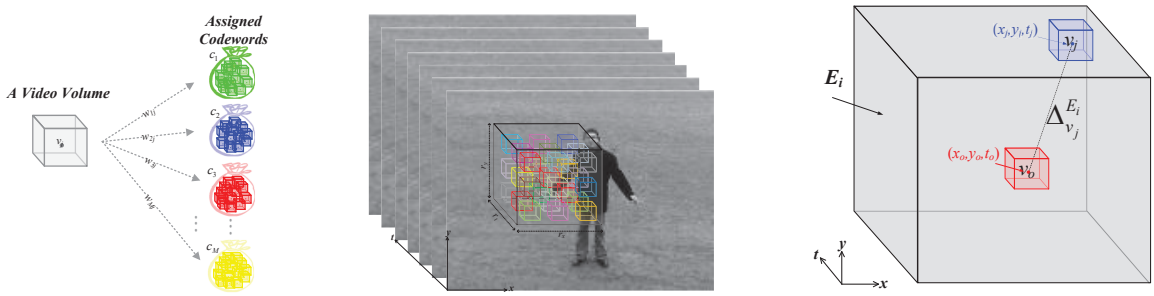
where $M(x,y,t)$ is the 3D gradient magnitude, and $\phi(x,y,t)$ and $\theta(x,y,t)$ are the orientations within $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$ and $[-\pi, \pi]$, respectively. The descriptor vector for each

video volume, taken as a histogram of oriented gradients (HOG), is constructed using the quantized θ and ϕ into n_θ and n_ϕ bins, respectively, weighted by the gradient magnitude, M. The descriptor of each video volume will be referred to as $h_i \in \mathbb{R}^{n_\theta+n_\phi}$. This descriptor represents both motion and appearance and possesses some degree of robustness to unimportant variations in the data, such as illumination changes [11, 97]. However, it should be noted that our algorithm does not rely on a specific descriptor for the video volumes, and other descriptors might enhance the performance of the approach. Examples of more complicated descriptors are the ones in [99] and in [49], the spatio-temporal gradient filters in [11, 132], the spatio temporal oriented energy measurements [129, 26] and the popular three-dimensional Scale Invariant Feature Transform (SIFT) [97].

2.3.1.2 *Codebook Of Video Volumes.* As the number of these volumes is extremely large (for example, about 10^6 in a one minute video) it is advantageous to group similar STVs to reduce the dimensions of the search space. This is commonly performed in all BOW approaches [62, 99]. Here, similar video volumes are also grouped when constructing a codebook. The procedure is straightforward and is described in Chapter 3. Thus, a normalized weight $w_{i,j}$ of assigning the codeword c_j to video volume v_i is given by (3.3)[2]. Eventually, each 3D volume, v_i, is assigned to the labels, c_j's, with a degree of similarity, $w_{i,j}$, as shown in Figure 2.2A. We note that the number of labels (shown in color), $M^{\mathcal{L}}$, is much smaller than the number of volumes, N. Moreover, codebook construction can be performed using any other clustering method, such as *k-means*, online fuzzy c-means [91], or mutual information [62].

2.3.2 High Level Scene Representations: Ensembles Of Visual Volumes. At the previous step, similar video volumes were grouped in order to construct the low-level codebook. The outcome of this is a set of similar volumes, clustered regardless of their positions in space and time. This is the point at which all other BOW methods in the literature stop. As stated earlier, the main drawback of

[2]Throughout the rest of the paper, each video volume will be represented by its descriptor vector.



(A) (B) (C)

FIGURE 2.2. (A) Codeword assignment to spatio-temporal video volumes. Each codeword is assigned to a volume with a degree of similarity $w_{i,j}$. (B) An ensemble of spatio-temporal volumes obtained at one of the computed scales. A large 3D volume surrounding each pixel, containing many spatio-temporal volumes, is considered and referred to as an ensemble of volumes. This large 3D volume will be used both for further analysis and measuring the likelihood of each pixel. (C) Relative spatio-temporal coordinates of a particular video volume inside an ensemble of volumes, $\Delta_{v_j}^{E_i}$.

many BOW approaches is that they do not consider the spatio-temporal composition (context) of the video volumes. Certain methods for capturing such information have appeared in the literature (see [14, 58, 66]). In this paper, we present a probabilistic framework for quantifying the arrangement of the spatio-temporal volumes.

2.3.2.1 *Ensembles Of Volumes.* Suppose a new video is to be analyzed; we refer to it as the *query*. The goal is to measure the likelihood of each pixel in the query video given a set of previously observed video for event description. To accomplish this, it is necessary to analyze the spatio-temporal arrangement of the volumes in the clusters that have been determined by the visual codebook. Thus, we next consider a large 3D volume around each pixel in (x, y, t) space. This large region contains many volumes with different spatial and temporal sizes as shown in Figure 2.2B. Thus it captures both the local and more distant information in the video frames. Such a set is called an *ensemble* of volumes around the particular pixel in the video. The ensemble of volumes, $E(x, y, t)$, surrounding each pixel (x, y) in the video at time t, is defined as:

$$E(x,y,t) = \left\{ v_j^{E(x,y,t)} \right\}_{j=1}^{J} \triangleq \left\{ v_j : v_j \subset R_{(x,y,t)} \right\}_{j=1}^{J} \tag{2.3}$$

where $R_{(x,y,t)} \in \mathbb{R}^3$ is a volume with pre-defined spatial and temporal dimensions centered at point (x, y, t) in the video (e.g., $r_x \times r_y \times r_t$) and J indicates the total number of volumes inside the ensemble. These large contextual 3D spaces are employed to construct higher-level codebooks.

2.3.2.2 *Contextual Information And Spatio-Temporal Compositions.* To capture the spatio-temporal compositions of the video volumes, we use the *relative* spatio-temporal coordinates of the volume in each ensemble, as shown in Figure 2.2C. Assume that the ensemble of video volumes at point (x_i, y_i, t_i) is E_i and the central video volume inside that ensemble is called v_o. Assume that v_o is located at the point (x_o, y_o, t_o) in absolute coordinates. Therefore, $\Delta_{v_j}^{E_i} \in \mathbb{R}^3$ is the relative position (in space and time) of the jth video volume, v_j, inside the ensemble of volumes:

$$\Delta_{v_j}^{E_i} = (x_j - x_o, y_j - y_o, t_j - t_o) \tag{2.4}$$

Then each ensemble of video volumes at point (x_i, y_i, t_i) is represented by a set of such video volumes and their relative positions, and hence (2.3) can be rewritten as:

$$E(x_i, y_i, t_i) = \left\{ \Delta_{v_j}^{E_i}, v_j, v_o \right\}_{j=1}^{J} \tag{2.5}$$

An ensemble of volumes is characterized by a set of video volumes, the central video volume, and the relative distance of each of the volumes in the ensemble to the central video volume, as represented in (2.5). This provides a view-based graphical spatio-temporal multi-scale description at each pixel in every frame of a video.

A common approach for calculating similarity between ensembles of volumes is to use the star graph model in [14, 75, 11]. This model uses the joint probability between a database and a query ensemble to decouple the similarity of the topologies of the ensembles and that of the actual video volumes [75]. To avoid such a decomposition, we estimate the *pdf* of the volume composition in an ensemble and then measure the similarity between these estimated *pdf*s.

During the codeword assignment process described in section 2.3.1.2, each volume v_j inside each ensemble was assigned to a label $c_m \in \mathbf{C}^{\mathcal{L}}$ with some degree of similarity

$w_{j,m}$. Given the codewords assigned to the video volumes, each ensemble of volumes can be represented by a set of codewords and their spatio-temporal relationships. Let $c_m \in \mathbf{C}^{\mathcal{L}}$ be the codeword assigned to the video volume v_j and $c_n \in \mathbf{C}^{\mathcal{L}}$, the codeword assigned to the central video volume v_o. Therefore, (2.5) can be rewritten as[3]:

$$\begin{aligned} v_j &\leftarrow c_m \\ v_o &\leftarrow c_n \\ E(x_i, y_i, t_i) &= \bigcup_{\substack{m=1:M^{\mathcal{L}} \\ n=1:M^{\mathcal{L}}}} \{\Delta, c_m, c_n\}_{j=1:J} \end{aligned} \tag{2.6}$$

where Δ denotes the relative position of the codeword c_m inside the ensemble of volumes. By representing an ensemble as a set of codewords and their spatio-temporal relationships, the topology of the ensemble, $\mathbf{\Gamma}$, is defined as:

$$\mathbf{\Gamma} = \bigcup_{\substack{m=1:M \\ n=1:M}} \{\Gamma_{m,n}(\Delta)\} \tag{2.7}$$

where $\mathbf{\Gamma}$ is the topology of an ensemble of video volumes that encodes the spatio-temporal relationships between codewords inside the ensemble. $\Gamma_{m,n}(\Delta) \in \mathbf{\Gamma}$ is taken to be the spatio-temporal relationship between two codewords, c_m and c_n in the ensemble[4]. Therefore,

$$\Gamma_{m,n}(\Delta) = (\Delta, c_m, c_n) \tag{2.8}$$

Let v denote an observation, which is taken as a video volume inside the ensemble. Assume that its relative location is represented by Δ_v, and v_o is the central volume of the ensemble. The aim is to measure the probability of observing a particular ensemble model . Therefore, given an observation, $\left(\Delta_{v_j}^{E_i}, v_j, v_o\right)$, the posterior probability of each topological model, $\Gamma_{m,n}$, is written as:

$$P\left(\Gamma_{m,n} | \left(\Delta_{v_j}^{E_i}, v_j, v_o\right)\right) = P\left(\Delta, c_m, c_n | \Delta_{v_j}^{E_i}, v_j, v_o\right) \tag{2.9}$$

[3] $\leftarrow$ symbolizes value assignment.

[4] These topological models, $\Gamma_{m,n}(\Delta)$, are obtained by assuming that the codeword entries are independent. Although in the case of overlapping video volumes such an assumption is not true, this is the standard Markovian assumption made for BOW.

The posterior probability in (2.9) defines the probability of observing the codewords c_m and c_n and their relative location, Δ, given the observed video volumes $\left(\Delta_{v_j}^{E_i}, v_j, v_o\right)$ in an ensemble of volumes. Equation (2.9) can be rewritten as:

$$P\left(\Delta, c_m, c_n | \Delta_{v_j}^{E_i}, v_j, v_o\right) = P\left(\Delta, c_n | c_m, \Delta_{v_j}^{E_i}, v_j, v_o\right) P\left(c_m | \Delta_{v_j}^{E_i}, v_j, v_o\right) \tag{2.10}$$

Since now the unknown video volume, v_j, has been replaced by a known interpretation, c_m, the first factor on the right hand side of (2.10) can be treated as being independent of v_j. Moreover, it is assumed that video volumes are independent. Thus v_o can be removed from the second factor on the right hand side of (2.10) and hence, it can be rewritten as follows:

$$P\left(\Delta, c_m, c_n | \Delta_{v_j}^{E_i}, v_j, v_o\right) = P\left(\Delta, c_n | c_m, \Delta_{v_j}^{E_i}, v_o\right) P\left(c_m | \Delta_{v_j}^{E_i}, v_j\right) \tag{2.11}$$

On the other hand, the codeword assigned to the video volume is independent of its position, $\Delta_{v_j}^{E_i}$. Therefore (2.11) can be reduced to:

$$P\left(\Delta, c_m, c_n | \Delta_{v_j}^{E_i}, v_j, v_o\right) = P\left(\Delta, c_n | c_m, \Delta_{v_j}^{E_i}, v_o\right) P\left(c_m | v_j\right) \tag{2.12}$$

Rewriting (2.12) gives:

$$P\left(\Delta, c_m, c_n | \Delta_{v_j}^{E_i}, v_j, v_o\right) = P\left(\Delta | c_m, c_n, \Delta_{v_j}^{E_i}, v_o\right) P\left(c_n | c_m, \Delta_{v_j}^{E_i}, v_o\right) P\left(c_m | v_j\right) \tag{2.13}$$

Similarly, by assuming independence between codewords and their locations, (2.13) can be reduced to:

$$P\left(\Delta, c_m, c_n | \Delta_{v_j}^{E_i}, v_j, v_o\right) = P\left(\Delta | c_m, c_n, \Delta_{v_j}^{E_i}\right) P\left(c_n | v_o\right) P\left(c_m | v_j\right) \tag{2.14}$$

The first factor on the right hand side of (2.14) is the probabilistic vote for a spatio-temporal position, given the codewords assigned to the central video volume of the ensemble, the codeword assigned to the video volume, and its relative position. We note that, given a set of ensembles of video volumes, the probability distribution function (*pdf*) in (2.14) can be formed using either a parametric model

or non-parametric estimation. $P(c_m|v_j)$ and $P(c_n|v_o)$ in (2.14) are the votes for each codeword entry and they are obtained in the codeword assignment procedure in section 2.3.1.2. Eventually, each ensemble of volumes can be represented by a set of *pdf*s as follows:

$$P\left(\mathbf{\Gamma}|E_i\right) = \bigcup_{\substack{m=1:M^{\mathcal{L}} \\ n=1:M^{\mathcal{L}}}} \left\{P\left(\Gamma_{m,n}\left(\Delta\right)|E_i\right)\right\} \tag{2.15}$$

where $P\left(\mathbf{\Gamma}|E_i\right)$ is a set of *pdf*s modeling topology of the ensemble of volumes. Therefore, once a video clip has been processed, each ensemble of spatio-temporal volumes has been represented by a set of *pdf*s as given in (2.15) and similarity between two video sequences can be computed simply by matching the *pdf*s of the ensembles of volumes at each pixel. In Chapters 3 and 4 we also show how those ensembles of volumes are employed to detect abnormal patterns in both space and time. Chapter 5 extends these concepts to the video-to-video matching problems and activity recognition.

2.4 Summary

In this chapter, an alternative approach for describing contextual information was presented. At first, the video is densely sampled and similar to the BOW structure, the first level codebook is formed. In order to improve the BOW structure, the concept of ensemble of volumes is introduced. Therefore, an ensemble of volumes is a large region around a particular pixel in the video which contains lots of STVs. To capture the contextual information, the spatio-temporal structure of the video volumes inside the ensembles is modeled using a probabilistic framework. The result of the processing in this chapter permits us to construct a set of local behavior patterns for each pixel based on the ensembles of volumes. This makes it possible to solve the following problems:

- Rare Event Detection: The ensembles of STVs are employed to compare a new observation to the previous observations. This will produce a self-similarity map of the video and rare events can be identified. In addition,

ensembles of STVs can be decomposed into two spatial- and temporal-oriented ensembles. This space/time decomposition makes it possible to identify pure spatial and temporal dominant/rare events (see Chapters 3 and 4).

- Video to Video Matching: The ensembles of video volumes can be used for constructing the second level codebook, called the high-level one or the bag of ensembles of volumes. Following the same inference mechanism in the traditional BOW, the activity recognition problem is solved which is described in Chapter 5.
- Tracking Moving Objects: Given the codebook of ensembles of volumes, the trajectories of moving objects are generated by linking the assigned codewords in consecutive frames (see Chapter 6).

Chapter 3

Abnormal Event Detection And Localization

3.1 Introduction

In light of the problem statements in Chapter 1, our goal is to build a fast anomaly detection framework for surveillance systems that addresses practical requirements, such as real-time performance and reliable detection and localization of anomalies. In addition, we seek the ability to learn newly observed events *without any offline and supervised training*. Perhaps most important, we will achieve this and not require any object tracking[1], background subtraction or other similar processes such as foreground segmentation methods with their attendant weaknesses, which form the basis for previously reported approaches.

The approach presented in this chapter focuses on the spatio-temporal abnormalities in the videos [90]. This is achieved by considering abnormality detection as a reconstruction problem. By formulating anomaly detection as a reconstruction process, anomaly detection is reduced to being defined as an outlier detection problem, i.e., finding the events that are not similar enough to the previously observed events in the video. Therefore, given a video sequence $\mathbf{V}$ containing a set of events $\mathbf{V} = \{e_i\}_{i=1}^{N}$ and a similarity measure S, the concept of an anomaly is defined for a

[1]Recall that at this point in the analysis that there has been no mention of object trajectories (see 1.2.2).

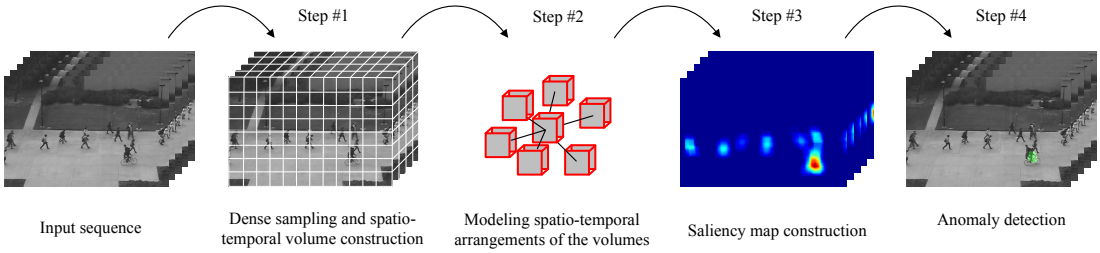


FIGURE 3.1. The goal is to detect anomalies in video data containing realistic scenarios. First, the video is densely sampled and spatio-temporal volumes are constructed at different spatial and temporal scales (#1). Then, similar spatio-temporal volumes are grouped and their spatio-temporal relationships modeled using a probabilistic framework (#2). Thus a video of salient events is constructed (#3) and the anomalous regions detected (#4) without the need for background subtraction and tracking.

particular event e_q, as follows:

$$\begin{aligned} & e_q \in \mathbf{V} \\ & s_{q,i} = S\left(e_q, e_i\right), \qquad e_i \in \mathbf{V} - \{e_q\} \\ & e_q \quad is \quad \text{anomaly} \quad iff \quad \forall i, \quad s_{q,i} \leq \gamma \end{aligned} \tag{3.1}$$

where γ is a threshold. This implies that the event e_q is not *similar enough* to any of the observed events. Given a short video clip containing valid behaviors, the method reconstructs newly observed behaviors using known examples of just valid ones. This is achieved by computing the likelihood of each pixel in each frame and eventually producing a likelihood or saliency map of all pixels in each frame. The likelihood map identifies the anomalous patterns which are inferred by selecting video arrangements with very low likelihood of occurrence. An overview of our approach is sketched in Figure 3.1.

After initialization of the algorithm using just a few seconds of video[2], our method builds a model of normal behavior in the form of codebooks and detects anomalies

[2]The number of initialization frames required to construct the ensemble of volumes (contextual region in the time domain) is related to the temporal size of the ensembles, R_{s_i,t_i}, (see (2.3)). We discuss the number of initialization frames in section 3.4.3.

while incrementally updating itself in an unsupervised manner when a new normal pattern is observed. The main characteristics of this approach are as follows:

- It introduces a probabilistic framework to capture spatio-temporal configurations of video volumes. This is achieved by estimating probability density functions of the *arrangements* of video volumes. Consequently, anomalies are defined as those spatio-temporal compositions in a video or set of videos having very low probability of occurrence.
- It significantly reduces the size of the database for finding *similar* examples to a new observation while retaining summary information, thereby speeding up the process and making it real-time.
- It uses an online unsupervised incremental method in order to update the probability distribution functions of the normal events. Thus, our method can adaptively learn newly observed normal patterns.

We have conducted extensive experiments to evaluate the capability of our approach for both anomaly detection and localization on different datasets with different normal/abnormal behavior patterns: anomalous walking patterns[3] [14]; the UCSD pedestrian dataset, which consists of two datasets[4]: UCSD Ped1 and UCSD ped2 [65]; subway surveillance videos[5] [2]; and an anomaly detection dataset[6] [129]. The results indicate that our approach is comparable to the state-of-the-art, while it can additionally be extended to more difficult problems[7].

3.2 Related Work

As indicated earlier, the recent trend in anomaly detection is to use spatio-temporal video volumes in the context of BOW models[8]. Their popularity is due

[3] http://www.wisdom.weizmann.ac.il/~vision/Irregularities.html

[4] http://www.svcl.ucsd.edu/projects/anomaly

[5] Obtained from the authors of [2]

[6] http://www.cse.yorku.ca/vision/research/spatiotemporal-anomalous-behavior.shtml

[7] All videos and additional results are available at: http://www.cim.mcgill.ca/~javan/index_files/Abnormal_events.html

[8] Essentially the probabilistic topic models, such as that of [133], can also be considered as BOW approaches since they ignore the spatio-temporal order of the local features [60].

to their low computational cost, as well as their ability to focus on abnormal behavior, even in extremely crowded scenes [56]. However, since classical BOV approaches group similar volumes, they destroy all compositional information in the process of grouping visual words [98, 58]. Thus, the likelihood of each video volume is based on its similarity to the other volumes in the dataset, without considering the spatio-temporal properties of neighboring ones. For example, in [2], motion patterns in local regions are estimated using optical flow and then quantized to construct a histogram of optical flow in local regions. Dissimilar motion patterns are considered to be anomalies.

It has been shown that anomaly detection by spatio-temporal volumes without considering their composition will produce unacceptable results [14, 34, 52, 54, 56, 58, 62, 75, 95, 116]. Several approaches have been presented to improve this situation. These are often based on co-occurrence matrices that are employed to describe contextual information. In large part, these methods are used exclusively for action recognition, since they require a supervised learning process. For example, the well-known correlogram exploits spatio-temporal co-occurrence patterns [95]. An alternative that does incorporate contextual information in a BOV framework is presented in [62], in which three-dimensional spatio-temporal pyramid matching is employed. This technique is based on the original two-dimensional spatial pyramid matching of multi-resolution histograms of patch features [58]. Likewise in [92], temporal relationships between clustered patches are modeled using ordinal criteria (e.g., equals, before, overlaps, during, after, etc.). In [34] the spatial information is coded through concatenation of video words detected in different spatial regions and data mining techniques to find frequently occurring combinations of features. Similarly, [66] addresses this issue by using the spatial configuration of the 2D patches by incorporating their weighted sum. In summary, most of these approaches are used for activity recognition rather than anomaly detection, and contextual information is represented locally and at fixed spatial or temporal scales.

In spite of the above, some efforts have recently been made to incorporate contextual information [11]. Here a local test for detecting abnormal video volumes measures the similarity of a particular video volume to its eight neighbouring volumes; those that are not similar to all others in this set are marked as being anomalous. In [56], video volumes are represented using 3D Gaussian distributions of the spatio-temporal gradient. The temporal relationship between these distributions is modeled using HMMs.

As indicated in Chapter 2 Boiman and Irani [14] have presented an alternative approach based on the spatio-temporal composition of a large number of volumes. Each new observation is *reconstructed* using only the previously observed spatio-temporal volumes, which are obtained by *densely sampling* the video. To consider the relationship between these volumes, the likelihood of a large contextual region around each volume is computed using the examples already seen in the video. By using densely sampled volumes, Boiman and Irani [14] were able to produce a good approximation to the likelihood, thereby permitting the detection of normal and abnormal behavior using a star graph model. The primary drawbacks of the work of Boiman and Irani [14] are the high computational complexity of their method and the lack of any means of taking into account uncertainty about the BOVs. We deal with both of these aspects by using a probabilistic framework to determine the likelihood of the space-time cuboids in a video. However, the main problem with dense sampling is its excessive computational time. Furthermore, it requires a large amount of memory to store all of the volumes as well as their spatio-temporal relationships.

Some modifications of [14] have been presented, but these are strictly within the framework of action recognition [88]. For example a modified version of [14] has been presented in [75], in which a two-level clustering method is employed to speed-up the search process. At the first level, all similar volumes are categorized. Then clustering is performed on randomly selected groups of spatio-temporal volumes while considering the relationships in space and time between the five nearest spatio-temporal volumes. However, the small number of spatio-temporal volumes involved

makes this method local in nature. Another hierarchical approach is presented in [54], which attempts to capture the compositional information of a subset of the most discriminative video volumes. We note that both of these approaches exploit supervised learning and hence cannot really be used for anomaly detection.

In order to avoid a computational bottleneck, we sort video volumes based on their spatio-temporal similarity, *while considering the uncertainty in the grouping procedure*. Although the method for clustering of similar video volumes may at first glance appear to be analogous to the Implicit Shape Model (IMS) in Leibe et al. for class-based object recognition [59], it actually differs in three respects. First, we examine the information in a large spatio-temporal context. Second, there is no need to create a predefined set of known object classes with predefined object centers. Our method adaptively determines these on its own. Third, we apply our method to videos, not just to images. By employing a probabilistic framework, the system we describe in this chapter significantly reduces the computational time required for determining the similarity of the compositions of the many video volumes that need to be examined [14]. Moreover, our approach also dramatically reduces the amount of memory required for storing previously observed video arrangements.

Thus the aim of this chapter is to demonstrate how videos can be processed for anomalies in real-time while summarizing the important information in the video. Ultimately, this will provide the ability to characterize and label all, not just anomalous, events. This is achieved by constructing a hierarchical BOW algorithm that learns both dominant and abnormal events in a unified framework. More closely related to our proposed approach are those methods that construct a spatio-temporal behavioral model of the scene [49, 8, 48, 30]. To date, these have focused on detecting low-level local anomalies in a video by analyzing the activity pattern of each pixel as a function of time. This activity pattern, also known as the busy/idle sequence of each pixel, is a binary sequence for each pixel in which 0s and 1s denote the foreground and background pixel in each frame, respectively. In [49], each pixel is processed independently and the relationships between the pixels in space and time are ignored,

thereby making such methods too local. In an improved version of [49], the spatial dependencies between pixels are taken as a function of pixel location by constructing a co-occurrence frequency matrix [8]. Although the latter has achieved good results for abnormality detection, the method requires that the activity pattern of each pixel be constructed by employing a conventional method for background subtraction. These are known to be deficient for non-stationary situations.

In contrast to the aforementioned approaches that attempt to model either local spatio-temporal activity patterns of a pixel or trajectories of moving objects, our goal is to construct a hierarchical model for all of the activities in a scene. We present a novel method for inference of motion patterns, which overcomes the drawbacks and limitations of the current methods, while employing simple yet powerful hierarchical methodologies.

3.3 Abnormal Event Detection

Figure 3.1 shows the steps of the proposed anomalous activity recognition algorithm, STC (Spatio-Temporal Compositions). At first, a codebook model is constructed to group similar spatio-temporal video volumes and remove redundant data; for example, in one minute of typical video, we have found experimentally that there are about 10^6 video volumes, while the number of codewords is around 20. Then, a large contextual region (in space and time) around each video volume is examined and the compositional relationships between video volumes are approximated using a mixture of Gaussians. To construct such a probabilistic model, a small number of video frames containing *normal* behaviors is necessary to initiate the on-line learning process. The minimum number of such frames is governed by the extent of the size of the temporal context[9]. Thus it is unnecessary to employ large numbers of training videos, containing valid behaviors, as is usually the case in the current literature.

The problem is transformed to a reconstruction problem using the previous formulation for anomaly detection (3.1). This equation implies that the similarity between

[9]We discuss the number of initialization frames in section 3.4.3.

a newly observed video frame and all previous observations is calculated according to (3.1). In order to make a decision about new observations in a reasonable amount of time, information regarding the spatio-temporal volumes and their relative arrangement in the regions of interest must be *efficiently* stored in the database. Here we focus on two issues, the *reconstruction process*, and a *fast inference mechanism* for anomaly detection. Therefore, the goal of the algorithm is to reduce the number of spatio-temporal volumes stored in the dataset in order to limit the search time, while still retaining a compact and accurate representation of the spatio-temporal arrangement of *all* volumes. As illustrated in Figure 3.1, the algorithm consists of two main steps: (1) sampling and coding the video to construct spatio-temporal volumes and probabilistic models of relative compositions of the spatio-temporal volumes, and (2) an inference mechanism to make decisions about newly observed videos. To construct such a probabilistic model for an arrangement of the spatio-temporal volumes of "normal" actions, it is necessary to use a few sample video frames containing such behaviors. These examples must be observed in order to initialize (or train) the algorithm. In the rest of this chapter, we refer to these video frames as the "training set". Although, currently, this probabilistic model is created during initialization, any other valid action that has not actually been observed during initialization can also be used.

3.3.1 Scene Modeling And Local Self-Similarity Maps. The essence of the method described in this section is to measure the similarity between various spatio-temporal volumes in the observation set and the incoming video data in order to examine whether the actions are anomalous. Thus, newly observed data must be re-constructed using historical data. In this section, we first explain the sampling strategy, followed by codebook construction for grouping similar video volumes. Then we describe the mechanism to capture spatio-temporal contextual information.

3.3.1.1 *The First Level Codebook.* Our work is based on the bag of features approach, i.e., a set of spatio-temporal volumes obtained using dense, random, or salient points. Although there are many methods for selecting the latter, dense sampling has

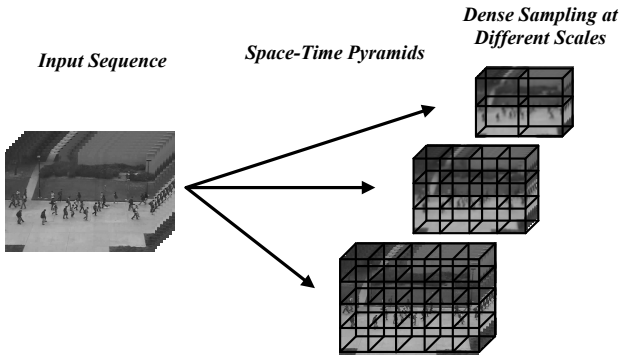


FIGURE 3.2. Dense sampling is performed at different spatial and temporal scales, producing a set of spatio-temporal volumes.

been shown to be superior to the others in terms of retaining the informative features of a video [84]. Therefore, performance almost always increases with the number of sampled spatio-temporal volumes, making dense sampling the preferable choice [111, 14].

Similar to the BOW structure described in the Chapter 2, the 3D spatio-temporal volumes in a video, $v_i \in \mathbb{R}^{n_x \times n_y \times n_t}$, are constructed by assuming a small volume of size of $n_x \times n_y \times n_t$ around *each* pixel in the video, in which $n_x \times n_y$ is the size of the spatial (image) window and n_t is the depth of the video volume in time. Spatio-temporal volume construction is performed at various spatial and temporal scales, producing a sort of video pyramid. Figure 3.2 illustrates the process of spatio-temporal volume construction.

Instead of using the HOG descriptor of Chapter 2, each spatio-temporal volume in the video is characterized by a set of simple descriptors as in [14, 27]. The descriptors are defined by the absolute value of the temporal derivatives of all pixels in each volume, v_i

$$\forall v_i, \qquad g_i = abs\left(\Delta_t\left(v_i\right)\right) \tag{3.2}$$

Their values are then stacked in a vector and normalized to a unit length to form a "compact" feature descriptor for each video volume at various scales, $h_i \in \mathbb{R}^{n_x n_y n_t}$. The procedure in (3.2) is actually performed at several spatial and temporal scales of

a Gaussian space-time video pyramid of the original data. In other words, the spatio-temporal volumes are smoothed at different scales before computing the gradients. An interesting property of this descriptor is that it is largely invariant to roughly static backgrounds, which makes it possible to detect abnormal actions regardless of the background, although only in surveillance systems in which the background does not change quickly. Notwithstanding its simplicity, the results obtained are very promising. Obviously the simple gradient descriptor defined in (3.2) could be replaced by others, and perhaps, thereby enhance the performance.

Here, similar spatio-temporal volumes are grouped by constructing a codebook, as detailed in Figure 3.3. This is a straightforward procedure. The first codeword is made equivalent to the first observed spatio-temporal volume. After that, by measuring the similarity between each observed volume and the codewords already in the codebook, either the codewords are updated or a new one is formed. Then, each codeword is updated with weight of $w_{i,j}$, which is based on the similarity between the volume and the existing codewords[10]. Here, we utilize the Euclidean distance for this purpose. Thus, the normalized weight of assigning codeword c_j to video volume v_i is given by[11]:

$$w_{i,j} = \frac{1}{\sum_{j} \frac{1}{distance(v_i, c_j)}} \times \frac{1}{distance\,(v_i, c_j)} \tag{3.3}$$

Another important parameter is the number of times that a codeword has been observed (f_j). The codebook is continuously being pruned to eliminate those that are either infrequent or very similar to the others, which ultimately generates M different codewords that are taken as the labels for the video volumes, $\mathbf{C} = \{c_i\}_{i=1}^{M}$. Since the goal of the algorithm is to measure similarity of a new observation to a subset of previously observed normal actions, the codebook is formed using videos that contain valid actions.

[10] Later it will be seen that this facilitates the handling of uncertainty in codeword assignment.

[11] Throughout the rest of this chapter, each video volume will be represented by its vector descriptor.

Consider that there are N video volumes in the dataset, represented by their descriptor vectors: $V = \{v_i\}_{i=1}^{N}$

Initialization

The first codeword is defined as:

- $c_1 \leftarrow v_1$
- $f_1 \leftarrow 1$
- $P_{l_1} \leftarrow 1$

Codebook Construction

- Construct a codeword using the Euclidean distance as the similarity measure between the volumes and codewords
- For all volumes (v_i) in the dataset,
  - If $\min_j distance\,(v_i, c_j) > \varepsilon$ construct a new codeword
    - $c_{j+1} \leftarrow v_i$
  - Else
    - Calculate $w_{i,j}$ using (3.3)
    - Update the codeword: $c_j \leftarrow \frac{f_j \times c_j + w_{i,j} \times v_i}{f_j + w_{icj}}$
    - Update the frequency: $f_j \leftarrow f_j + w_{icj}$
    - Prior probability of the codeword: $P\,(c_j) = \frac{f_j}{N}$

Pruning the Codebook

For all codewords, $\{c_m\}_{m=1}^{M}$

- If $\{distance\,(c_i, c_j) < \alpha \times \varepsilon,\ (0 < \alpha < 1)\}$ *and* $\left\{f_j < 0.1 \times \frac{N}{M}\right\}$
  - Merge the two codewords
    - Remove codewords c_i and c_j from the codebook
    - Define the new codeword as: $c_{M+1} \leftarrow \frac{f_i \times c_i + f_j \times c_j}{f_i + f_j}$
    - The corresponding frequency of the new codeword: $f_{M+1} \leftarrow f_i + f_j$

FIGURE 3.3. Codebook construction and pruning procedure

After the initial codebook formation[12], each 3D volume, v_i, can be assigned to all labels, c_j's, with a degree of similarity, $w_{i,j}$, as shown in Figure 2.2A. The codebook construction can be performed using any other clustering method, such as *k-means* or mutual information [62]. In section 4.3 we replace this codebook formation method by an online version of fuzzy c-means clustering algorithm.

3.3.1.2 *Capturing The Topology Of The Ensembles of STVs.* As discussed in Chapter 2, the main drawback of most BOW approaches is that they do not consider the spatio-temporal context of each volume. Thus the outcome is a set of similar volumes, clustered regardless of their positions in space and time. Several methods for capturing such information have appeared in the literature (see [14, 58, 66]). Here,

[12]Recall that initialization requires a minimum of one video frame.

we present an alternative probabilistic framework for quantifying the arrangement of the spatio-temporal volumes at a pixel in the video.

Consider a new visual observation, the *query*. The goal is to estimate the likelihood of each pixel in the query being normal. To accomplish this, a large region R around each pixel is considered and the likelihood is calculated by *measuring the similarity between the volume arrangement in the query* and the dataset as described by (3.1). Given the representation of an ensemble of volumes in (2.3), abnormality detection is reduced to constructing a similarity map of new observations with respect to *all of the previous ones*. In doing this, the similarity between many different topologies of ensembles of volumes will be taken into account in order to capture the specific context of each pixel. The use of the spatio-temporal context surrounding a pixel will tend to influence the ultimate choice of code word associated with a particular pixel.

Let us consider how we represent an ensemble of video volumes, E_i, at (x_i, y_i, t_i) containing K spatio-temporal volumes. Thus the ensemble, E_i, is centered at a video volume v_i located at the point (x_i, y_i, t_i) in absolute coordinates. Here we use the *relative spatio-temporal coordinates* of the volume in an ensemble to account for its position, as shown in Figure 3.4A. Consider the kth volume in E_i. Define $\Delta_{v_k}^{E_i} \in \mathbb{R}^3$ as the relative position (in space and time) of the kth video volume, v_k, located at the point (x_k, y_k, t_k), inside the ensemble of volumes. $\Delta_{v_k}^{E_i}$ is defined by(3.4):

$$\Delta_{vk}^{E_i} = (x_k - x_i, y_k - y_i, t_k - t_i) \tag{3.4}$$

Then each ensemble of video volumes at location (x_i, y_i, t_i) is represented by a set of such video volumes and their relative positions with respect to the central video volume. Hence (2.3) can be rewritten as:

$$E_i = \left\{\Delta_{v_k}^{E_i}, v_k, v_i\right\}_{k=1}^{K} \tag{3.5}$$

where K is the total number of video volumes inside the ensemble.

FIGURE 3.4. (A) Relative spatio-temporal coordinates of a particular video volume inside an ensemble of volumes. (B) Codeword assignment to the video volumes inside the ensemble E_i, and their relative distance in the codeword space. $c \in \mathbf{C}$ represents the codeword assigned to the kth video volume inside the ensemble, v_k. $c' \in \mathbf{C}$ denotes the codeword assigned to the central video volume of the ensemble, v_i. The random variable δ represents the relative position of these codewords in the codeword space (see text).

During the codeword assignment process described in the section 3.3.1.1, a codeword $c \in \mathbf{C}$ was assigned to each video volume, v_k, inside each ensemble with an associated degree of similarity using (3.3). Given the codewords assigned to the video volumes, each ensemble of volumes can be represented by a set of codewords and their spatio-temporal relationships. Assume that $\mathcal{V} \subset \mathbb{R}^{n_x n_y n_t}$ is the space of the descriptors for a video volume, and $\mathbf{C}$ is the codebook constructed in section 3.3.1.1. Let $c: \mathcal{V} \to \mathbf{C}$ be a random variable, which assigns a codeword to a video volume. Assume that $c': \mathcal{V} \to \mathbf{C}$ is a random variable denoting the assigned codeword to the central video volume of an ensemble. Therefore, $\delta: \mathbb{R}^3 \to \mathbb{R}^3$ is a random variable denoting the relative position of a codeword c to the codeword assigned to the central video volume of the ensemble, c'. Given the above assumptions, an ensemble of volumes can be represented as a graph of codewords and their spatio-temporal relationship, as shown in Figure 3.4B.

Having defined the representation of an ensemble of volumes in (3.5), and given the assigned codewords to the video volumes as described above, a set of hypotheses describing the topology of each ensembles can be defined. Those hypotheses are then used for constructing a similarity map between the topologies of the ensembles in

a new observation with respect to all of the previous ones. Let us consider each hypothesis, $\mathbf{h}$, as a tuple $\mathbf{h} = (c, c', \delta)$. Therefore, the set of hypotheses, $\mathcal{H}$[13], which describe the topology of each ensemble is defined as follows:

$$\mathcal{H} = \bigcup_{\mathbf{h}} \{\mathbf{h}\} = \bigcup_{\substack{c \in \mathbf{C} \\ c' \in \mathbf{C}}} \{(c, c', \delta)\} \tag{3.6}$$

Suppose we now consider sampling the video frame-by-frame and pixel-by-pixel in each frame. Let $\mathcal{O} = (v_k, v_i, \Delta_{v_k}^{E_i})$ signify a single observation, where v_k denotes any observed video volume inside an ensemble, E_i; v_i denotes the observed video volume at the center of the ensemble; and $\Delta_{v_k}^{E_i}$ is the relative location of the observed video volume, v_k, with respect to the v_i inside E_i. The aim is to measure the probability of each hypothesis given the observation. Therefore, given an observation, $\mathcal{O}$, the posterior probability of each hypothesis, h, is written as:

$$P\left(\mathbf{h} \mid \mathcal{O}\right) = P(c, c', \delta \mid v_k, v_i, \Delta_{v_k}^{E_i}) \tag{3.7}$$

The posterior probability in (3.7) defines the probability of observing the codewords c, c', and their relative position, δ, given the observed video volumes, $\left(v_k, v_i, \Delta_{v_k}^{E_i}\right)$. Then (3.7) can be rewritten as:

$$P(c, c', \delta \mid v_k, v_i, \Delta_{v_k}^{E_i}) = P(c', \delta \mid c, v_k, v_i, \Delta_{v_k}^{E_i}) P(c \mid v_k, v_i, \Delta_{v_k}^{E_i}) \tag{3.8}$$

Since an observed video volume, v_k, has been replaced by a postulated interpretation, c, the first factor on the right hand side of (3.8) can be treated as being independent of v_k. Moreover, it is assumed that video volumes v_k and v_i are independent[14]. Hence, v_i can be removed from the second factor on the right hand side of (3.8). Therefore (3.8) can be rewritten as:

$$P(c, c', \delta \mid v_k, v_i, \Delta_{v_k}^{E_i}) = P(c', \delta \mid c, v_i, \Delta_{v_k}^{E_i}) P(c \mid v_k, \Delta_{v_k}^{E_i}) \tag{3.9}$$

[13] These hypotheses, $\mathcal{H}$, are obtained by assuming that the codeword entries are independent.

[14] Although in the case of overlapping video volumes such an assumption is not true, this is the standard Markovian assumption made for BOV.

On the other hand, the codeword assigned to a video volume is independent of its position, $\Delta_{v_k}^{E_i}$. Therefore (3.9) can be reduced to:

$$P(c, c', \delta \mid v_k, v_i, \Delta_{v_k}^{E_i}) = P(c', \delta \mid c, v_i, \Delta_{v_k}^{E_i}) P(c \mid v_k) \tag{3.10}$$

so that rewriting (3.10) gives:

$$\begin{aligned} P(c, c', \delta \mid v_k, v_i, \Delta_{v_k}^{E_i}) &= P(c', \delta \mid c, v_i, \Delta_{v_k}^{E_i}) P(c \mid v_k) \\ &= P(\delta \mid c, c', v_i, \Delta_{v_k}^{E_i}) P(c' \mid c, v_i, \Delta_{v_k}^{E_i}) P(c \mid v_k) \end{aligned} \tag{3.11}$$

Similarly, by assuming independence between codewords and their locations, (3.11) can be reduced to:

$$P(c, c', \delta \mid v_k, v_i, \Delta_{v_k}^{E_i}) = P(\delta \mid c, c', \Delta_{v_k}^{E_i}) P(c' \mid v_i) P(c \mid v_k) \tag{3.12}$$

Knowing the codeword assigned to the video volume, c, and the codeword assigned to the central video volume of the ensemble, c', the first factor on the right hand side of (3.12), $P(\delta \mid c, c', \Delta_{v_k}^{E_i})$, is the probabilistic vote for a spatio-temporal position, δ. Thus, given a set of ensembles of video volumes, it can be formed using either a parametric model or non-parametric estimation. Here, we approximate this *pdf* using a mixture of Gaussians. The maximum number of Gaussians is set to three and the parameters of the Gaussians are optimized using an expectation-maximization procedure [28]. The second and third terms in the right hand side of (3.12), $P(c' \mid v_i)$ and $P(c \mid v_k)$, are the votes for each codeword entry and are obtained as a result of the codeword assignment procedure[15].

Thus, given an ensemble of spatio-temporal video volumes, the likelihood of its composition can be computed simply by using the *pdf*s instead of laboriously comparing all other video volumes compositions in the dataset. As discussed in the next section, anomalous events are determined from these *pdf*s by selecting those compositions with very low likelihood of occurrence. Comparing this with [14], in which an exhaustive search was employed to determine the optimal ensemble, our approach

[15] Codewords are assigned to the video volumes regardless of their location in space and time.

is capable of retaining adequate information about the spatio-temporal arrangement of the volumes while reducing the memory requirements. It also greatly reduces the dimension of the search space for finding similar regions in the dataset for a new observation.

3.3.2 Detecting Anomalous Patterns: Inference Mechanism. Next, consider the scenario of a continuously operating surveillance system. At each temporal sample t, a single image is added to the already observed frames and the resulting video sequence, the *query*, $\mathcal{Q}$, is formed. In order to detect anomalous patterns, the posterior probability of each pixel in the query video is calculated using the ensemble of the spatio-temporal volumes around it to determine whether the point is related to the normal events or is suspicious.

Given (3.6) which details the ensemble topology hypotheses, $\mathcal{H}$ obtained from the previous section, the posterior probability of an ensemble of volumes in the query is calculated as: $P\left(\mathcal{H} \mid E_i^{\mathcal{Q}}\right)$. Here $E_i^{\mathcal{Q}}$ is an ensemble of video volumes in the query centered at point (x_i, y_i, t_i).

Thus given $E_i^{\mathcal{Q}}$, we wish to search for previously observed ensembles that are most similar to the newly observed ensemble in terms of both their video volumes and topologies. In other words, the posterior probability should be maximized:

$$\max_{\mathbf{h}} P\left(\mathcal{H} \mid E_i^{\mathcal{Q}}\right) = \max_{\substack{c \in \mathbf{C} \\ c' \in \mathbf{C}}} P\left(c, c', \delta \mid E_i^{\mathcal{Q}}\right) \tag{3.13}$$

Since we represent each ensemble by its spatio-temporal video volumes, relative position and the central volume, and assuming that the observed video volumes are independent[16], the right side of the above equation can be written as the product of the posterior probability of every video volume inside the ensemble:

$$P\left(c, c', \delta \mid E_i^{\mathcal{Q}}\right) = \prod_{k}^{K} P\left(c, c', \delta \mid q_k, q_i, \Delta_{q_k}^{E_i^{\mathcal{Q}}}\right) \tag{3.14}$$

[16] This is the Markov assumption (see [14, 75]).

Inference mechanism for a new query video, $\mathcal{Q}$

Sampling and Coding

- Construct space-time pyramids for $\mathcal{Q}$
- Densely sample the video at all scales and construct spatio-temporal volumes: $\{q_1, q_2, ..., q_K\}$
- Codeword assignment:
  - Assign each q_k in the query to the obtained codewords $\{c_1, c_2, ..., c_M\}$ with a similarity $w_{k,m}$ using the same *distance* function used during the training process as described in Figure 3.3
- Construct ensembles of spatio-temporal volumes, $E_i^{\mathcal{Q}}$

Similarity Map Construction

For each ensemble of volumes, $E_i^{\mathcal{Q}}$ containing K spatio-temporal volumes:

- For each volume q_k inside $E_i^{\mathcal{Q}}$, compute the relative position using (3.4) $\Delta_{q_k}^{E_i^{\mathcal{Q}}} \quad k = 1 : K$
- Calculate the probability of the ensemble $E_i^{\mathcal{Q}}$ being normal: $S_{E_i^{\mathcal{Q}}} = \max_{\substack{c \in \mathbf{C} \\ c' \in \mathbf{C}}} P\left(c, c', \delta \mid E_i^{\mathcal{Q}}\right)$

Decision-making regarding the observation

Given the calculated similarity of each ensemble of volumes, $S_{E_i^{\mathcal{Q}}}$

$E_i^{\mathcal{Q}} \quad is \quad anomaly \quad if \quad S_{E_i^{\mathcal{Q}}} \leq \gamma$

FIGURE 3.5. Anomalous action detection (Inference mechanism).

where q_k is the video volume inside $E_i^{\mathcal{Q}}$, q_i is the central volume of $E_i^{\mathcal{Q}}$, $\Delta_{q_k}^{E_i^{\mathcal{Q}}}$ is the relative position of the q_k, and K is the total number of spatio-temporal video volumes inside the ensemble. Referring to (3.12), it is obvious that $P\left(c, c', \delta \mid q_k, q_i, \Delta_{q_k}^{E_i^{\mathcal{Q}}}\right)$ in (3.14) can be rewritten as follows:

$$P(c, c', \delta \mid E_i^{\mathcal{Q}}) = \prod_{k}^{K} P(\delta \mid c, c', \Delta_{q_k}^{E_i^{\mathcal{Q}}}) P(c \mid q_k) P(c' \mid q_i) \tag{3.15}$$

Thus the maximum posterior probability in (3.13) can be rewritten as:

$$\max_{\substack{c \in \mathbf{C} \\ c' \in \mathbf{C}}} P\left(c, c', \delta \mid E_i^{\mathcal{Q}}\right) = \max_{\substack{c \in \mathbf{C} \\ c' \in \mathbf{C}}} \prod_{k}^{K} P(\delta \mid c, c', \Delta_{q_k}^{E_i^{\mathcal{Q}}}) P(c \mid q_k) P(c' \mid q_o) \tag{3.16}$$

This is a straightforward computation because the prior probability of each spatio-temporal volume in the query has been calculated during codeword assignment (described in section 3.3.1.1). The posterior probability is calculated using the estimated probability distribution functions in section 3.3.1.2. Figure 3.5 shows the pseudo-code for the inference process.

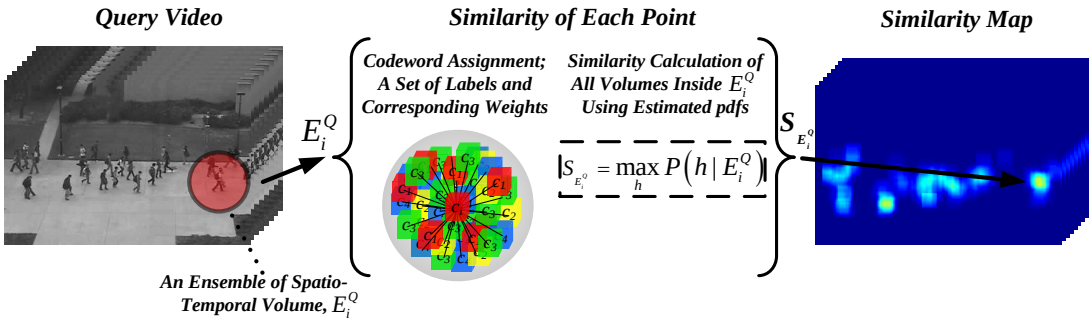


FIGURE 3.6. Likelihood map construction for each pixel in the video frame. The query video is densely sampled and the likelihood of each pixel at different spatial and temporal scales is computed within a large region around it. $E_i^{\mathcal{Q}}$ contains many spatio-temporal volumes. This data structure facilitates the computation of the similarity between all volumes and their local context. The computation involves both new and previously observed data. The likelihood of each point is calculated using the probabilistic model of the volume arrangements and a likelihood map of the whole frame is constructed. Inferring the location of the anomalous regions is accomplished by thresholding the likelihood map.

In summary, at first, the query, $\mathcal{Q}$ is densely sampled at different spatio-temporal scales in order to construct the video volumes. Each volume q_k is assigned to a codeword $c \in \mathbf{C}$ with similarity obtained from (3.3). The probability of being normal of every pixel in a video frame is then calculated using the spatio-temporal arrangement of the volumes inside each ensemble, $E_i^{\mathcal{Q}}$. Ultimately, the likelihoods of each pixel in the video frame will yield a *similarity map of the whole frame*. As a result, the likelihood of every pixel in each frame is approximated (see Figure 3.6). Clearly, the regions in a frame of the video containing suspicious behaviors will have less similarity to the examples already existing in the dataset. Thus, decisions about anomalous actions can be made using the calculated similarity map, which is based on a threshold. In the experiments described in section 3.4, a *single threshold* for all test sequences was applied to the similarity map. The similarity map was processed before thresholding by a spatio-temporal median filter to reduce noise effects and outliers.

We also note that the proposed statistical model of codeword assignment and the arrangement of the spatio-temporal volumes permit small local misalignments in

the relative geometric arrangement of the composition. This property, in addition to the multi-scale volume construction in each ensemble, enables the algorithm to handle certain non-rigid deformations in space and time. This, of course, is necessary since human actions are not exactly reproducible, even for the same person. We conclude this section by examining computational complexity. Suppose there are K video volumes available in each ensemble and the number of codewords is M. For each ensemble, the time complexity of the codeword assignment is $O(K \times M)$ and for the maximum posterior probability in (3.16) is $O(K \times M \times M)$. Thus, the inference mechanism for each ensemble of video volumes in the query has the time complexity of $O(K \times M \times (M+1))$. On the other hand, the method proposed by Boiman and Irani [14], which is the exact solution to anomaly detection by reconstruction, has a time complexity of $O(K \times N)$, in which N is the total number of video volumes previously observed. Moreover, in [14] N video volumes are required to be stored in memory as previous observations, while in our approach the total number video volumes stored is M. Noting that usually $M << N$, the approach proposed here requires much fewer computations as well as a smaller amount of memory space.

3.3.3 Algorithm Initialization. Before continuing with the experimental results, we describe how the algorithm is initialized. The scenario we have considered here implies on-line and continuous surveillance of a particular scene in order to detect anomalous patterns. Therefore, we require only that the first N frames of the video stream initiate the process. Furthermore, N should be taken at least equal to the temporal size of the ensembles in order to construct a successful model of the previous observations. These N frames must contain only normal events, and we have referred to them as the training or initialization sequence. The actual number of initialization frames (N) required and its effect on the detection results is discussed in the next section. To initiate the codebook during the first N frames, each video volume is assigned to a codeword with a similarity weight using the procedure explained in section 3.3.1.1. In addition, probability distribution functions of spatio-temporal arrangements of the codewords are also estimated. This can be accomplished either

online or offline. When the next frame, $(N+1)$th frame, arrives it is densely sampled to construct spatio-temporal video volumes and the ensembles of these video volumes. Their similarity to the volumes that have already been obtained is computed using the codebook constructed during the initialization procedure and inference mechanism described in section 3.3.2. In this way, the algorithm constantly learns newly observed normal events in an unsupervised manner (see experimental results). Similar to [2, 14], dominant events are assumed to be normal while rarely observed activities are considered as anomalies.

3.4 Experiments

The algorithm was tested on crowded and non-crowded scenes (one or two persons in the scene) in order to measure the capabilities of the proposed method for anomalous activity recognition. Four publicly available datasets of anomalous events were used: the anomalous walking patterns of a person[17] [14]; the UCSD pedestrian dataset, which has recently been published and actually consists of two datasets[18] [65]; the subway surveillance videos[19] [2]; and the anomaly detection dataset[20] [129], the last containing videos captured under variable illumination conditions. Except for the first dataset, the others were gathered in realistic environments. To evaluate performance, we also compared the results with other pixel-level approaches of current interest, such as Inference by Composition (IBC) [14], Mixture of Dynamic Textures (MDT) [65], Space-Time Markov Random Fields (ST-MRF) [52], Local Optical Flows [2], and spatio-temporal oriented energy filters [129][21]. The IBC method is currently considered to be one of the most accurate for pixel level saliency detection[22] and was

[17] http://www.wisdom.weizmann.ac.il/~vision/Irregularities.html

[18] http://www.svcl.ucsd.edu/projects/anomaly

[19] Obtained from the authors of [2].

[20] http://www.cse.yorku.ca/vision/research/spatiotemporal-anomalous-behavior.shtml

[21] Note computer code for these methods is not available publicly and had to be programmed using just the original papers as references.

[22] Our experimental results also support this claim.

tested to demonstrate that our proposed method (Spatio-Temporal Compositions, STC) produced similar results.

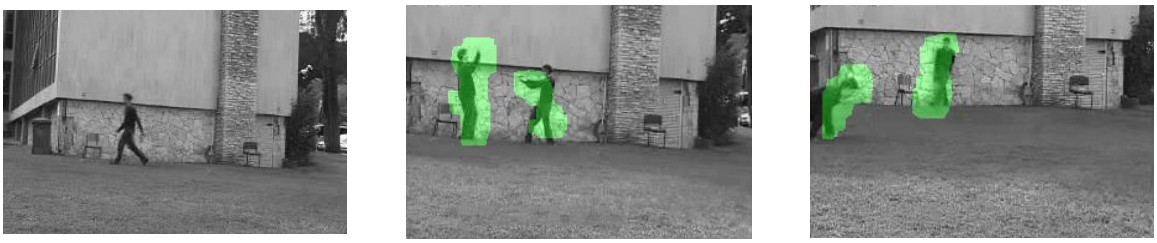
IBC calculates the likelihood of *every point* in each frame. This is achieved by examining the spatio-temporal volumes and their arrangements in a large region surrounding the pixels in a query video. ST-MRF models the normal activity using multiple probabilistic PCA models of local optical flow [52], while MDT can be considered as an extension of the dynamic texture-based model and is capable of detecting both spatial and temporal abnormalities [65]. Although the latter requires a large training dataset, it was used here for comparing results because of its superior performance on the UCSD pedestrian dataset.

3.4.1 Datasets For Anomaly Detection. The first dataset we discuss illustrates the situation with one or two persons in the scene. The *training*[23] video is short (24 seconds) and contains normal acted behaviors representing two different actions, walking and jogging by a single person. Figure 3.7 shows some sample images from this training set. The *query* is a long video clip which contains both acted normal and abnormal behaviors of one or two persons in the scene. In some sequences one of them performs a normal and the other, a suspicious action. The existence of the *simultaneous* occurrence of both normal and suspicious activities in the video provides an opportunity to evaluate the localization ability of the proposed method. The suspicious behaviors in the dataset are abnormal walking patterns, crawling, jumping over objects, falling down, etc. We show some frames in which the proposed algorithm detected suspicious behaviors in Figure 3.7[24].

The second dataset used for performance evaluation of the proposed approach is the UCSD pedestrian dataset. It contains video sequences from two pedestrian walkways where abnormal events occur. The dataset contains different crowd densities,

[23] Although our method does not actually require any specific number of training images, the training sequences specified for each dataset in the literature description of the experiments were used as the initialization frames.

[24] The videos showing results of our algorithm for abnormality detection are available at: http://www.cim.mcgill.ca/~javan/index_files/Abnormal_events.html



(A) (B) (C)

FIGURE 3.7. Detecting suspicious behavior in a scenario in which walking and jogging are the *normal* behaviors. (A) Valid example of walking. (B), (C) Abnormality detection in the query video by our proposed algorithm. New valid behaviors are automatically inferred from the dataset (e.g., two different persons walking and jogging). The anomalous regions are those that cannot be reconstructed using the example in (A) and these regions have been highlighted in green. In (B), the man holding a gun at the right and the one holding his hands up at the left are showing suspicious behaviors. In (C), the person at the left is crawling while the other is walking in an abnormal fashion.

and the anomalous patterns are the presence of non-pedestrians on a walkway (bicyclists, skaters, small carts, and people in wheelchairs). The UCSD pedestrian dataset contains 34 normal video clips for the first scene (UCSD Ped 1) and 36 video clips containing one or more anomalies for testing; and 16 normal video clips for the second scene (UCSD Ped 2), together with 14 test video clips. Figure 3.8 shows samples of these two scenes with the suspicious regions labeled by the proposed method.

The third dataset contains two actual surveillance videos of a subway station [2] recorded by a camera at the entrance and exit gates. The entrance gate surveillance video is 96 minutes long. It shows normal events such as going down through the turnstiles and entering the platform. There are also scenes containing 66 anomalous events, mainly walking in the wrong direction, irregular interactions between people and some other events, including sudden stopping, running fast, etc. [2]. The second one, the exit gate surveillance video, is 43 minutes long and contains 19 anomalous events, mainly walking in the wrong direction and loitering near the exit [2]. Neither the surveillance videos nor groups of frames within them are labelled as training or

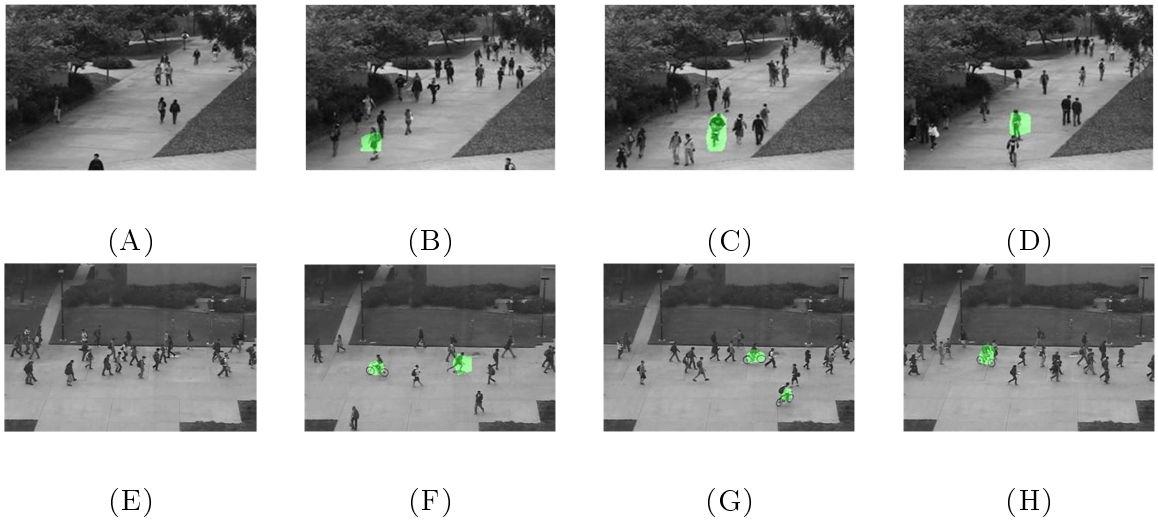


FIGURE 3.8. Detecting abnormalities using the UCSD pedestrian datasets. (A), (E) Sample frames of normal actions for the two scenes, containing only walking pedestrians. (B), (C), (D); (F), (G), (H) Abnormality detection in the query videos are highlighted in green. The bikers and the skater are the detected anomalous patterns.

testing videos. Figure 3.9 shows some frames from this dataset together with the detected anomalies using our approach.

The fourth dataset contains real-world videos with more complicated dynamic backgrounds plus variable illumination conditions. Notwithstanding the significant environmental changes in this dataset, the abnormalities are actually simplistic motions (e.g., motion in the scene or different motion direction). We used three videos from this dataset, which have variable illumination and dynamic backgrounds: the *Train*, the *Belleview*, and the *Boat-Sea* video sequences. The *Train* sequence is the most challenging one in this dataset [129] due to drastically varying illumination and camera jitter. In this sequence, the abnormalities relate to the movement of people. The other sequence is a traffic scene in which the lighting conditions change gradually during different times of the day and the abnormalities are cars entering the intersection from the left or right. In the last video sequence the abnormalities are the passing boats in the sea. Similar to the subway surveillance video dataset, there

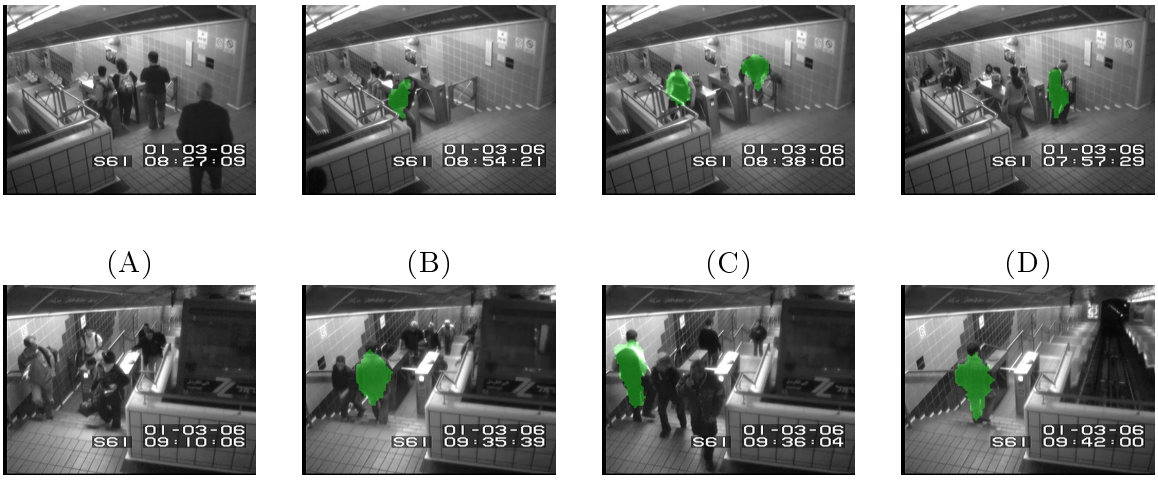


FIGURE 3.9. Anomaly detection at subway entrance and exit gates. The top row shows the entrance and the bottom shows the exit gate. Anomalous regions are highlighted in green. (A), (E) Show sample frames of two scenes. The anomalies are shown as follows: (B), (C) a person is exiting through the entrance gate; (F), (G), (H) a person is entering through the exit gate; (D) Two persons are trying to pass through the entrance gate without payment.

are no separate training and testing sequences. Figure 3.10 shows some frames of this dataset together with the detected anomalies using our approach.

3.4.2 Performance Evaluation: Abnormality Detection And Localization. Performance evaluation of any anomaly detection method can be conducted either at the frame or pixel level. Frame level detection implies that a frame is marked as suspicious if it contains *any* abnormal pixel, regardless of its location. On the other hand, pixel level detection attempts to measure the localization ability of an algorithm. This requires the detected pixels in each video frame to be compared to a pixel level ground truth map. Clearly, such abnormality localization is more important than marking the whole frame as suspicious.

We first consider a quantitative comparison of different approaches for anomaly detection at the frame level. Figure 3.11 shows the receiver operating characteristic (ROC) for the first dataset (containing anomalous walking patterns), plotted as a function of the detection threshold for different anomaly detection methods. Following

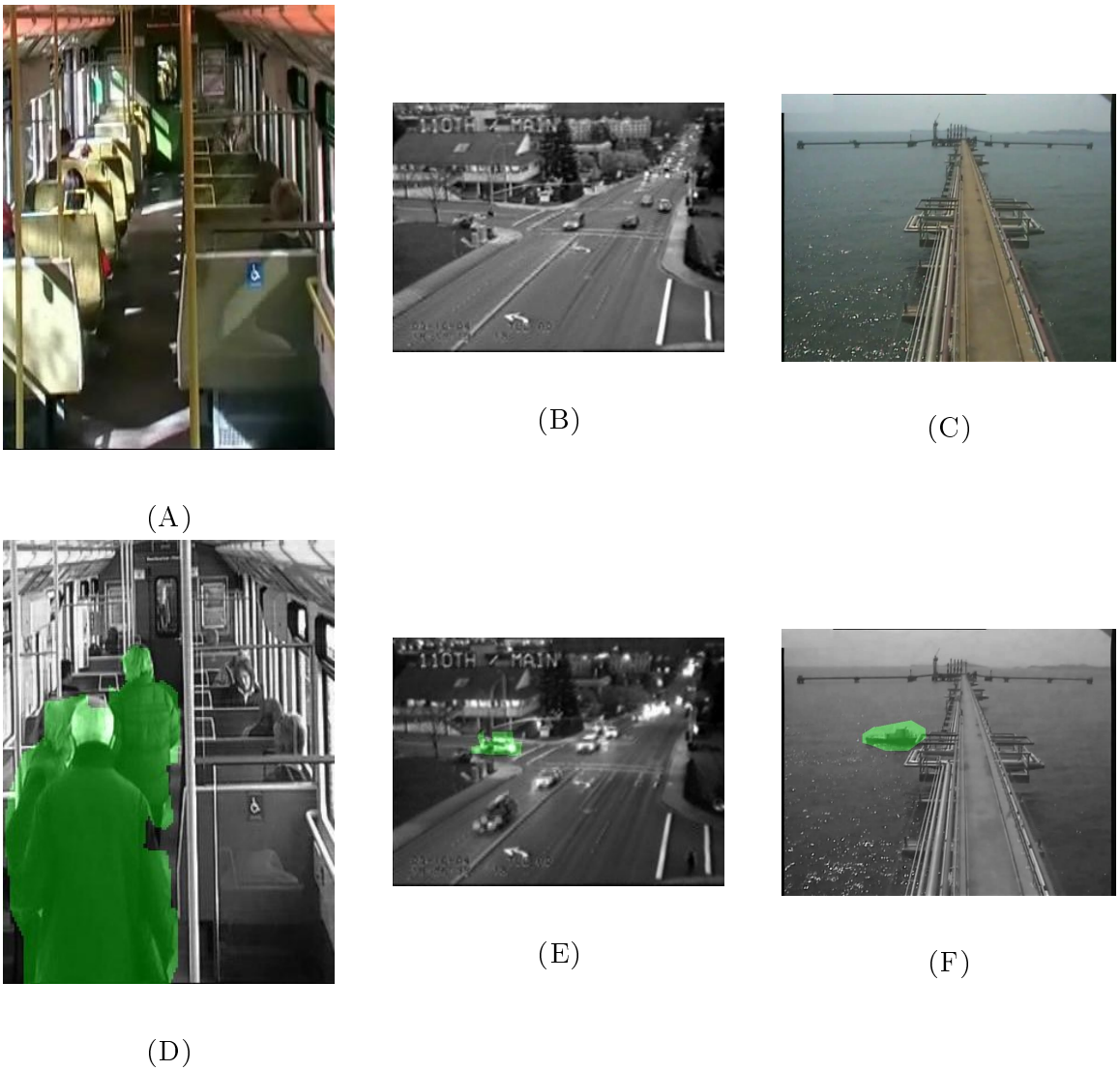


FIGURE 3.10. Anomaly detection in challenging datasets. The top row shows the sample frames from three datasets: (A) *Train*, (B) *Belleview*, and (C) *Boat-Sea* video sequence. The bottom row shows anomalous regions highlighted in green. Detected anomalous regions: (D) moving person, (E) Van detected moving to the right, (F) Boat fading out under a bridge.

the evaluation procedure of [2] and [14], each frame is marked as abnormal if it contains at least one pixel detected as an anomaly. Similarly we performed frame level detection on the UCSD pedestrian dataset and the ROC curves are illustrated in Figure 3.12A and 3.12B. It is clear from Figures 3.11 and 3.12 that IBC and STC produce more accurate results than the others, particularly MDT on the UCSD

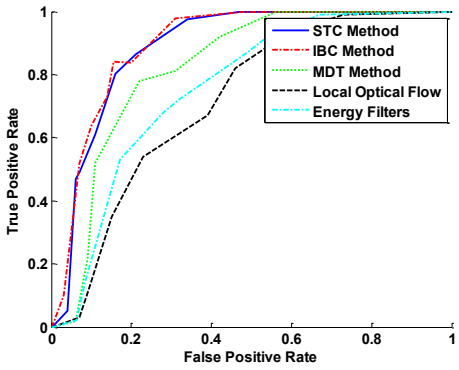


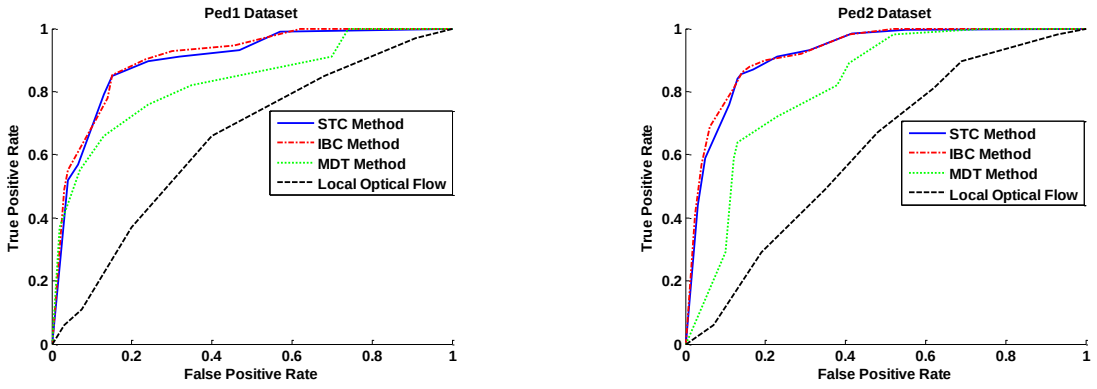
FIGURE 3.11. A comparison of the ROCs for the following methods: the proposed STC, IBC, MDT, local optical flow, and spatio-temporal oriented energy filters computed for the first dataset, which deals with anomalous walking patterns. ROC curves were obtained by varying the detection threshold, γ, for STC, the threshold on the overall abnormality map in MDT, and the saliency map in IBC.

pedestrian dataset. We note that MDT had been reported to have achieved the highest recognition rate for the UCSD dataset [11]). We observe that the similar performance of STC and IBC was probably predictable, because STC summarizes the spatio-temporal relationships between the video patches, while IBC maintains these by storing *all* spatio-temporal arrangements of *all* volumes in the dataset. This indicates that there was *no performance loss*, notwithstanding the fact that STC is based on probabilities and performs in real-time. Thus while the two methods may achieve similar results for anomalous event detection, our approach has two main advantages over IBC. First it is considerably faster (see Table 3.1) and, second, it requires much less memory to store the learned data. These issues would also be important if our approach were to be used to describe and summarize normal rather than just anomalous behaviors.

The second approach for performance evaluation is to measure the localization performance by evaluating it at the pixel level. To date, pixel level localization can

[25] Although there are very fast methods for local abnormality detection, such as [129], we compare the computational time of the algorithms for non-local and complicated behavior understanding that accounts for contextual information as well.

[26] This is the approximate computational time obtained on a PC with an Intel Q9550 CPU and 4GB of RAM.



(A) (B)

FIGURE 3.12. Comparing ROC of the proposed STC, IBC, MDT and the local optical flow method over two datasets. (A), (B) Performance based on the UCSD pedestrian datasets (UCSD ped 1 and USCD ped 2). Among all of these approaches, the proposed STC method shows similar results to IBC [14] and outperforms the others. However, compared to IBC, our method is much faster and requires much less memory.

TABLE 3.1. Required computational time for the tested methods for non-local abnormality detection using different datasets[25].

Processing time per frame (sec)[26]			
Dataset	Algorithm used for anomaly detection		
	STC Method	MDT Method	IBC Method
Ped1	0.19	21	69
Ped2	0.22	29	83
Subway Surveillance Videos	0.24	38	113
Walking Patterns	0.23	32	74

only be measured for a small number of datasets among existing public databases, since it requires ground truth maps. USCD pedestrian datasets [65], and the anomaly detection dataset [129] are the two datasets that include ground truth maps in which each region containing an anomalous event is marked manually. Thus the detected pixels in each video frame are compared to the ground truth map at the pixel level. For UCSD pedestrian datasets, anomaly detection is deemed to have occurred when at least 40% of the actual anomalous pixels have been detected[27]. Otherwise it is

[27] We used a single threshold for all videos in this study to mark suspicious regions. However, these are usually larger or smaller than the actual ground truth region. A degree of overlap of 40%

TABLE 3.2. Quantitative comparison of the proposed method (STC) and the state-of-the-art for anomaly detection using the UCSD pedestrians dataset. (* indicates that the method is claimed to have real time performance).

Algorithm used for anomaly detection	Dataset	EER for frame level detection	EER for pixel level detection	Number of frames containing valid examples (training sequences)
*STC	UCSD Ped1	15%	27%	200
	UCSD Ped2	13%	26%	180
MDT (Mahadevan et. al, 2010, [65])	UCSD Ped1	25%	58%	6800
	UCSD Ped2	24%	54%	2880
IBC (Boiman and Irani, 2007, [14])	UCSD Ped1	14%	26%	6800
	UCSD Ped2	13%	26%	2880
*Zaharescu and Wildes, 2010, [129]	UCSD Ped1	29 %	41%	6800
	UCSD Ped2	27%	36%	2880
*Bertini et. al, 2012, [11]	UCSD Ped1	31%	70%	6800
	UCSD Ped2	30%	-	2880
*Reddy et.al, 2011, [85]	UCSD Ped1	22.5%	32%	6800
	UCSD Ped2	20%	-	2880
Antic and Ommer, 2011, [5]	UCSD Ped1	18%	-	6800
	UCSD Ped2	14%	-	2880
ST-MRF (Kim and Grauman, 2009 [52])	UCSD Ped1	40%	82%	6800
	UCSD Ped2	30%	-	2880
*Local optical flow (Adam et. al, 2008 [2])	UCSD Ped1	38%	76%	6800
	UCSD Ped2	42%	-	2880

considered to be a false alarm. The equal error rate (EER), the percentage of misclassified frames when the false positive rate is equal to the miss rate, is calculated for both pixel and frame level analyses and presented in Table 3.2.

The results in Table 3.2 demonstrate that the proposed method (and, of course, IBC) outperformed other approaches both at the frame and pixel levels. Furthermore, it can detect anomalous patterns without significant performance degradation when there is perspective distortion and changes in spatial scale (UCSD Ped 1 dataset). This is in distinction to optical flow approaches that cannot handle this issue easily [65]. Moreover the computational time required by the method described in this chapter is significantly lower than other non local-approaches in the literature. In order to make a fair comparison of different approaches, the STC algorithm must

is a typical overlapping ratio suggested as the standard protocol for UCSD pedestrian dataset [65] and also used in [11, 85, 5] for measuring the localization ability of the algorithms.

be judged against other real-time algorithms as indicated in Table 3.2. Thus, we observe that the STC algorithm outperforms all other real-time algorithms and achieves the best results for the UCSD pedestrian dataset at both frame level detection and pixel level localization. It should also be noted that the results reported in Table 3.2 for all other methods were obtained using 50 video sequences for training (6800 video frames), while our approach used just one short video sequence consisting of 200 frames. This is a major advantage of our algorithm, which does not require long video sequences for initialization. It is also interesting to note that among other approaches that do not account for spatio-temporal contextual information, spatio-temporal oriented energy filters [129] is the fastest and outperforms other local approaches with real-time performance.

We also carried out experiments on another real-world video dataset, the subway surveillance dataset. The training strategy for the subway surveillance video is different from the UCSD pedestrian dataset, since no training set containing only normal events is available. Therefore, we used two approaches for initialization. The first one exploited a fixed number of frames, which is similar to previously reported approaches. Analogous to [52] and [131], we picked the first 5 minutes of the entrance gate video and the first 15 minutes of the exit gate video for initialization. The second approach was to continue learning newly observed events while still detecting the anomalies. The results are presented in Table 3.3. Compared with the other approaches for abnormality detection, the STC algorithm produces comparable results to the state of the art. We also observe that that performance of our algorithm is independent of the initialization strategy, although continuous learning does provide slightly better results.

We also evaluated the localization performance of our algorithm using pixel ground truth. Abnormality detection was performed for the subway exit gate video using the same initialization strategy as the frame level detection. The ground truth

[28] Used a reduced number of abnormalities, as their method was unable to detect all kinds of abnormalities in the dataset.

[29] Ibid.

TABLE 3.3. Comparison of different methods and learning approaches for the subway videos. In the fourth column, the first number denotes the detected anomalous events; the second is the actual number of anomalous events. (* indicates that the method is claimed to have real time performance)

Algorithm used for anomaly detection	Dataset	Training period	Number of anomalous events	False alarm
*STC	Entrance gate	5 min.	60/66	4
	Exit gate	15 min.	19/19	2
*STC	Entrance gate	Continuous	61/66	4
	Exit gate	Continuous	19/19	2
ST-MRF(Kim and Grauman, 2009 [52])	Entrance gate	5 min.	57/66	6
	Exit gate	15 min.	19/19	3
*Dynamic Sparse Coding (Zhao et. al, 2011 [131])	Entrance gate	Continuous	60/66	5
	Exit gate	Continuous	19/19	2
Sparse reconstruction (Cong et. al, 2011 [24][28])	Entrance gate	10 min.	27/31	4
	Exit gate	10 min.	9/9	0
*Local optical flow (Adam et. al, 2008 [2][29])	Entrance gate	5 min.	17/21	4
	Exit gate	15 min.	9/9	2

map for this video was produced manually by the authors of [129] for wrong way motion abnormalities. Figure 3.13 illustrates the precision-recall curves of the proposed algorithm and that of the spatio-temporal oriented energies method [129]. The method presented in this chapter shows superior performance. We attribute this to the fact that it accounts for contextual information in the scene and hence, it is capable of learning complicated behaviors. Although adding contextual information increases the computational complexity of the STC algorithm when compared to local approaches, it is still fast enough for real-time abnormality detection and localization.

Although the experiments described above indicate that our method can detect complicated abnormal behaviors in realistic scenes (UCSD pedestrian dataset and subway surveillance videos), we also conducted experiments for the fourth dataset. Although this dataset contains relatively simple abnormal events, we tested it to evaluate the effect of continuous learning under variable and difficult illumination conditions. We followed the same strategy for initialization of the algorithm as in [129], in which the first 800 frames of the *Train* video and the first 200 frames of

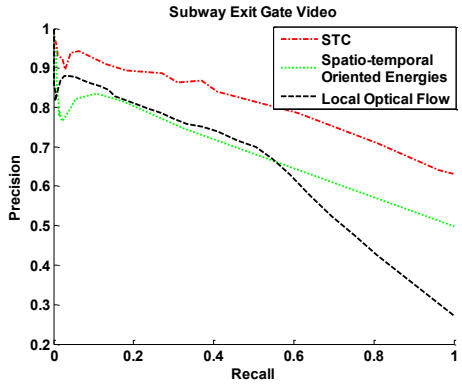
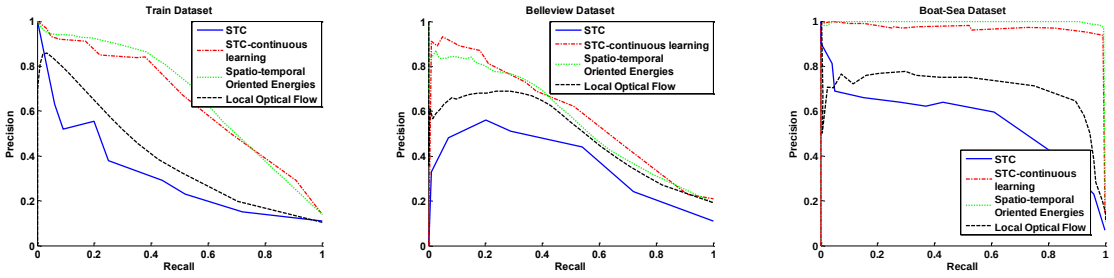


FIGURE 3.13. Comparing precision/recall curves for abnormality localization in the subway exit gate video surveillance sequence.



(A) (B) (C)

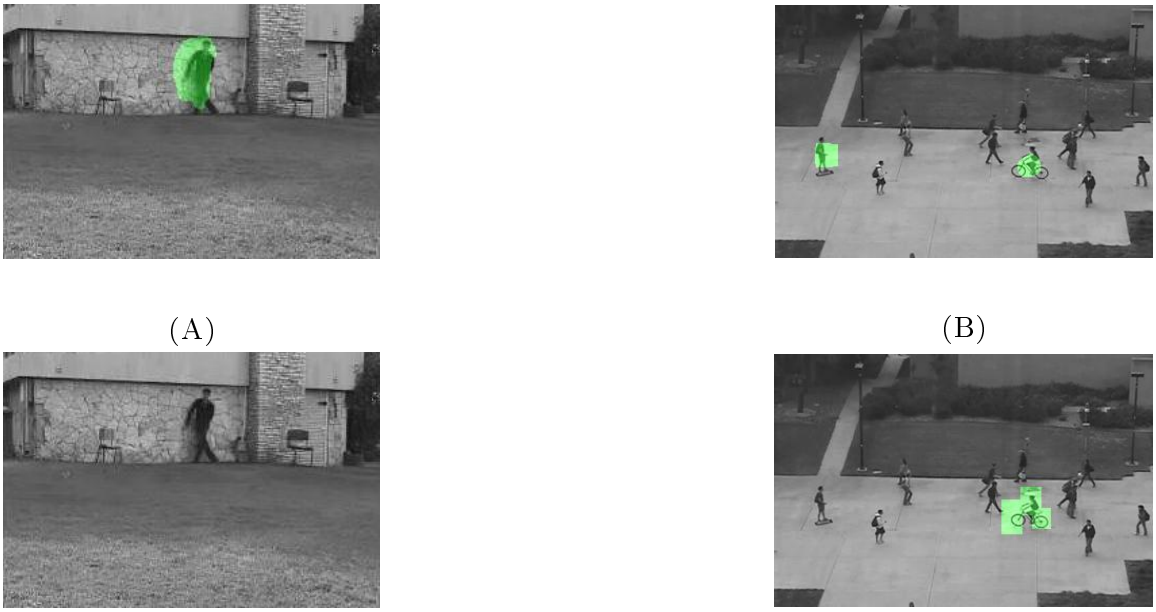
FIGURE 3.14. Comparing precision/recall curves for two videos of a challenging dataset: (A) *Train* video sequence, in which the illumination conditions change drastically in a short period of time, (B) the *Belleview* traffic scene, in which the lighting conditions change gradually from daylight to night, and (C) the *Boat-Sea* video sequence in which the background shows quasi-periodic patterns.

the *Belleview* and *Boat-Sea* video sequences were considered to be the initialization frames (these contain a total of 19218, 2918, and 2207 frames, respectively). We compared the results with two alternative pixel-level anomaly detection methods: spatio-temporal oriented energies [129] and local optical flow [2]. Although the abnormalities in this dataset are actually low level motions, we exclude pixel-level background models and behavior template approaches [49] from our comparisons as they do not achieve acceptable results [129]. The precision-recall curve of the STC method and two alternatives are presented in Figure 3.14.

Comparing first the performance in Figure 3.14 of the two strategies (red and blue curves) employed by STC, it is obvious that using *simultaneous* and *continuous* learning and detection of abnormalities (red curve) is superior to employing only an initial training set (blue curve). On the other hand, we observe that simple local optical flow features, combined with online learning [2] (black curve), do not yield acceptable results in the former case. Notwithstanding this, we also note that [2] was actually fairly capable of detecting abnormalities in other realistic datasets (Tables 3.2 and 3.3). Therefore, it appears that the optical flow approach (black curve) has difficulty capturing temporal flicker and dynamic textures. In the case of rapid changes in illumination, using a more complex feature descriptor, such as oriented energies (green curve) in [129], produces slightly better results than STC (the *Train* sequence) with faster execution time. On the other hand, we stress that this method cannot be used for more complex behaviors for two reasons: it is too local and does not consider contextual information[30]. Figure 3.15 illustrates two examples of abnormal behaviors in which this method fails.

3.4.3 Performance Evaluation: Effect of codebook size and number of initialization frames. As STC creates a codebook to group similar video volumes, it is necessary to analyze the effect of different codebook sizes on the performance of the algorithm. This is achieved by changing the threshold, ϵ, during codeword formation (see section 3.3.1.1). Various threshold values were used and the EER calculated. In Figure 3.16A, the EER value for frame level detection is plotted as a function of the codebook size (number of codewords) for the UCSD pedestrian dataset. We observe that large threshold values produce small codebooks, resulting in inadvertent merges of video volumes. This means that some local information may be lost, and furthermore, anomalous events may be grouped with the normal ones. On the other hand, as the number of the codewords increases, the algorithm stores more volumes, and in the extreme case, would be similar to the original IBC method. Using larger codebooks demands more memory and dramatically increases computational time, so

[30] Experimental results presented in Table 3.2 and Figure 3.13 also supports this claim.



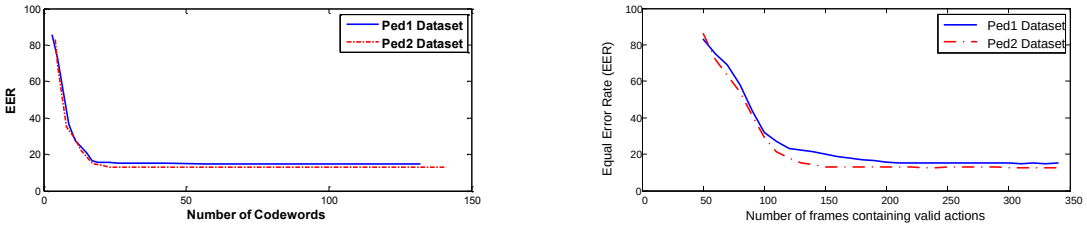
(A) (B)

(C) (D)

FIGURE 3.15. Comparing STC and spatio-temporal oriented energy methods on two datasets with more complicated abnormal behaviors: (A), (B) Results of STC for the Walking pattern and UCSD ped2 datasets. (C), (D) results of spatio-temporal oriented energy on these datasets.

that online implementation would become impossible. Although there is a trade-off between codeword size and the performance of the algorithm, it can be inferred from our experiments that using relatively small codebooks (e.g., 20 codewords) achieves acceptable results for anomaly detection.

Another major concern for learning algorithms in videos surveillance systems is the size of the training set, that is, how many valid examples are necessary for anomaly detection in a new video. We have tested this for STC using videos containing valid behaviors. The video size ranged from short sequences of 50 frames to longer ones containing 400 frames. Figure 3.16B shows the learning curve for UCSD Ped 1 and Ped 2 based on the EER. We observe that convergence is very fast. Therefore the proposed method is capable of detecting suspicious actions by observing just a few valid behaviors ($\sim$ 150 frames). Since, as indicated in Table 3.1, the present version



(A) (B)

FIGURE 3.16. Effect of codebook size and number of initialization frames on the STC algorithm for anomaly detection. The EER is calculated for frame level detection using UCSD Ped 1 and Ped 2 datasets. (A) Effect of codebook size on anomaly detection. (B) Learning curve of the STC method for anomalous action detection.

of the program runs at about 4-5 frames per second, we can infer that initialization requires about 20 seconds for this dataset.

3.5 Summary

The results presented in section 3.4 indicate that the STC method has a competitive performance (in terms of accuracy and computational cost) compared to the other approaches for anomaly detection for four challenging datasets. Moreover, it is fast enough for online applications and requires fewer initialization frames. When a separate training set is not available, the algorithm is capable of continuously learning the dominant behavior in an unsupervised manner while simultaneously detecting anomalous patterns. Clearly, this is the preferred behavior for any potential visual surveillance system operating in an unconstrained environment.

Overall, the STC algorithm produces similar results to the state-of-the-art for complicated abnormality patterns, while the computational cost is much lower. On the other hand, for local abnormalities, such as changes in background and temporal flicker in a scene with a complicated background, the method presented in [129] appears to be faster than STC, but weaker at dealing with the non-local patterns created by anomalous behaviors.

The main advantage of STC is that it takes into account the compositional information of the video volumes in a large region. Notwithstanding the simple temporal difference feature used here to describe the video volumes in STC, the algorithm is still capable of handling significant illumination variations. Thus using more complicated ones, most likely would further enhance the outcome. Although our results indicate that the STC method has competitive performance compared to other approaches, it still yields some errors. Analyzing these indicates that occlusion is the major source of error in crowded scenes. This was predictable as the video data were obtained using a single camera. Based on the experiments, we can summarize the results of our study as follows:

(i) In the case of *complicated* abnormal behaviors without drastic changes in illumination or dynamic backgrounds (in Walking patterns, UCSD pedestrian and Subway surveillance datasets):
   (a) STC outperforms all other *realtime* and *non-realtime* methods (except IBC) in terms of abnormality detection and localization.
   (b) STC produces similar results to IBC with vastly fewer computations.

(ii) In the case of *simple* abnormal events (motion/direction detection in the fourth dataset) with dynamic backgrounds and variable illumination conditions:
   (a) Continuous learning makes STC capable of handling environmental changes. Moreover, it is more robust to gradual changes, as it requires updating the *pdf*s to learn newly observed behaviors.
   (b) For drastically changing background and illumination, spatio-temporal oriented energy filters [129], which is dedicated to pixel level motion and direction detection, achieved better results than STC.

Chapter 4

Online Dominant And Anomalous Event Modeling

4.1 Introduction

In light of the problem statements in Chapter 1, our goal is to build a fast system that recognizes abnormal patters. In Chapter 3 we have described an automated system for abnormality detection. The proposed approach was based on constructing a self-similarity map of the video to identify spatio-temporal abnormal events. In this chapter, we address the problem of *simultaneously learning dominant and rare events in space and time*. This problem is a generalized problem of abnormality detection, in which a model is learned for dominant events. In addition, spatio-temporal events are decomposed into spatial and temporal events to capture abnormalities in both space and time.

Here we seek to simultaneously parse an *entire video* into local spatio-temporal regions in order to detect *all activities, anomalies and objects* using *unsupervised learning*. In addition, we will show that this can be achieved using a single unified formalism without possessing any models of the contents beforehand[91]. Figure 4.1 illustrates an example of video parsing. Normal events observed in a scene will be referred to as the "dominant" behavior. These are events that have a higher probability of occurrence than others in the video and hence generally do not attract much

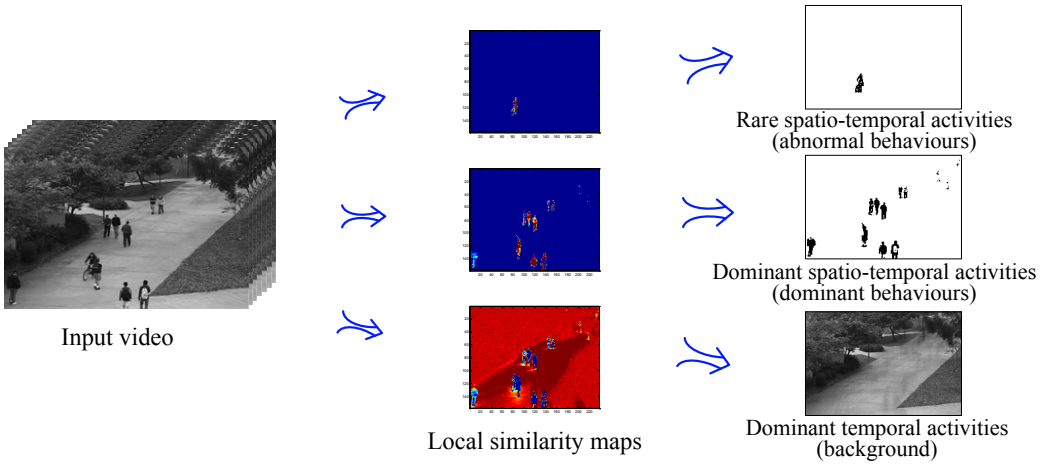


FIGURE 4.1. Video parsing. The input video is parsed into three meaningful components: background, dominant activities (walking pedestrians), and rare activities (the bicyclist).

attention. We can further categorize the dominant behavior into two classes. In the literature on attention, the first one usually deals with foreground activities in space and time [9, 8, 30, 49, 48] while the other describes the scene background[1]. Typically, the latter is more restrictively referred to as background subtraction, which is the building block of almost all computer vision algorithms. However, dominant behavior detection is more general and more complicated than background subtraction, since it includes the scene background while not being limited to it. The manner in which these two differ is the way that they use the scene information. Most background subtraction methods are based on the principle that the photometric properties of the scene in the video, such as luminance and color, are stationary. In contrast, dominant behavior understanding can be seen as a generalization of the classical background subtraction method in which all of the dynamic content (background and foreground) of the video come into play. Thus, we define these terms in a different manner from the current literature in that the background is taken as being spatial or temporal aspects that are normally occurring in the scene.

[1]By definition, the background consists of pixels in the video frames whose photometric properties, such as luminance and color, are either static or stationary with respect to time.

The main challenge is to learn both dominant and anomalous behaviors in videos of different spatio-temporal complexity. For example, these could range from non-stationary scene backgrounds to abnormal human activities. By achieving this, it becomes possible to construct a hierarchical layered model of the scene to understand the different behaviors. Thus, the algorithm can simultaneously model *high level behaviors* and detect *abnormalities* by considering both spatial and temporal contextual information while also performing *temporal pixel level change detection* and background subtraction. This characteristic makes our algorithm more general than both abnormality detection and background subtraction methods on their own. More precisely, the main characteristic of our approach are as follows: *I*- The spatio-temporal contextual information in a scene is decomposed into separate spatial and temporal *contexts*, which make the algorithm capable of detecting purely spatial or temporal activities, as well as spatio-temporal abnormalities. *II*- High level activity modeling and low level pixel change detection are performed simultaneously by a single algorithm. Thus the computational cost is reduced since the need for a separate background subtraction algorithm is eliminated. This makes the algorithm capable of understanding behaviors of different complexity. *III*- The algorithm adaptively learns the behavior patterns in the scene in an online manner. As such, the approach is a preferable choice for visual surveillance systems. *IV*- The major benefit of the algorithm is its *extendibility*, which is achieved by hierarchical clustering. This makes the algorithm capable of understanding dominant behaviors of different complexity.

In order to evaluate capabilities of our approach for dominant behavior understanding and abnormality detection, we have conducted experiments using different datasets with different dominant behavior patterns. The results indicate that our approach is comparable to the state-of-the-art, while it can be extended to more difficult problems[2].

[2]All videos and additional results are available at: http://www.cim.mcgill.ca/~javan/index_files/Dominant_behavior.html

4.2 Related Work

The problem of dominant behavior detection addressed in this paper can be considered either as a generalization of background subtraction [53, 134], scene understanding [108] or the inverse problem of abnormality detection [2, 14, 11, 129]. We limit the discussion of the literature to methods that are not based on a priori models but rather learn the dominant and abnormal behaviors.

To date, most of the reported approaches for behavior understanding that are not based on a priori models are grounded on trajectory analysis of the objects, which requires precise tracking methods [72, 76]. This remains a challenge, particularly in complex situations. On the other hand, techniques that do not require object detection followed by tracking focus on *local* spatio-temporal behaviors in videos and have recently gained increased popularity [8, 41]. Most of these methods rely mainly on extracting and analyzing low-level visual features, such as color, motion and texture in local regions in space and time. This is achieved either by constructing a pixel-level background model and behavior template [53, 49, 9, 30, 70] or by employing spatio-temporal video volumes [14, 11, 52, 133]. In large part, the former relies on an analysis of the activity pattern of each pixel in each frame as a function of time, i.e. the background subtraction process. These are used to construct a background model by mainly analyzing the photometric features at each pixel over time. More advanced approaches also incorporate the spatio-temporal compositions of the motion-informative regions to build background and behavior templates [9, 70] that are subtracted from newly observed behaviors in order to detect an anomaly. A review of the background subtraction method can be found in [78] and more recent work is presented in [7].

As indicated in previous chapters, the recent trend in video analysis (including dominant behavior understanding, scene understanding, abnormality detection and also human action recognition) is to use spatio-temporal video volumes in the context of BOW models. The classical BOW and probabilistic topic models often ignore the spatio-temporal relationships between video volumes. However, this is crucial

for accurate scene understanding [88, 86]. Although there have been some efforts to incorporate either spatial or temporal compositions of the video volumes into the probabilistic topic models, they suffer from high computational complexity. Therefore, they cannot be employed for online behavior understanding and real-time scene monitoring [42].

More closely related to our proposed approach are those methods that construct a spatio-temporal behavioral model of the scene [49, 8, 48, 30]. To date, these have focussed on detecting low-level local anomalies in a video by analyzing the activity pattern of each pixel as a function of time. This activity pattern, also known as the busy/idle sequence of each pixel, is a binary sequence for each pixel in which 0s and 1s denote the foreground and background pixel in each frame, respectively. In [49], each pixel is processed independently and the relationships between the pixels in space and time are ignored, thereby making such methods too local. In an improved version of [49], the spatial dependencies between pixels are taken as a function of pixel location by constructing a co-occurrence frequency matrix [8]. Although the latter has achieved good results for abnormality detection, the method requires that the activity pattern of each pixel be constructed by employing a conventional method for background subtraction. These are known to be deficient for non-stationary situations.

In contrast to the aforementioned approaches that attempt to model either local spatio-temporal activity patterns of a pixel or trajectories of moving objects, our goal is to construct a hierarchical model for all of the activities in a scene. We present a novel method for inference of motion patterns, which overcomes the drawbacks and limitations of the current methods, while employing simple yet powerful hierarchical methodologies.

4.3 Simultaneous Dominant and Rare Event Modeling

Here we concentrate on detecting two of the elements in Figure 4.1, that is, dominant spatio-temporal activities and abnormal behavior in a video. We focus on

low-level visual features, and begin by proposing a set of large contextual regions containing many of these features as well as their compositional information. Therefore, a simple and effective method for learning dominant behaviors as well as the low-level events in a dynamic scene is constructed. To accomplish this, we create a framework, which considers both the hierarchical nature of dominant behaviors, as well as their spatio-temporal context. As opposed to trajectory-based methods for behavior understanding [72, 76], our approach is grounded on a pixel-by-pixel analysis. Using *densely sampled* spatio-temporal video volumes (STVs), we create both local and global compositional graphs of volumes at each pixel. Although employing STVs in the context of bag of video words (BOV) has been extensively studied for the well-known problem of activity recognition, generally it involves supervised training. Here we do not use any training sets at all but continuously update time-varying BOV lookup tables. Therefore, our approach has the ability to learn newly observed behaviors without any offline or supervised training. After initializing the algorithm, typically using one or two seconds of video, the system builds an adaptive model of the dominant behavior while simultaneously detecting anomalies.

Consider the structure of the algorithm in Figure 4.2. Initially, the video is densely sampled, STVs are constructed, and similar ones are grouped to reduce the dimensions of the search space. Codebook construction of STVs is performed in an online manner while considering uncertainties in the codeword assignment. Then, a large contextual region containing many STVs (in space and time) around each pixel is examined and the compositional relationships between STVs are approximated using a probabilistic framework. We are interested in detecting different kinds of behavior in the spatial and temporal domains. To achieve this, we cluster all of the STVs that constitute all of the compositional graphs obtained during a time period in the near past. We use a modified version of online fuzzy clustering and thereby track the dominant spatio-temporal activities (clusters). These clusters of STVs provide concurrent distinctive spatial and temporal *models* of the scene. For example, we

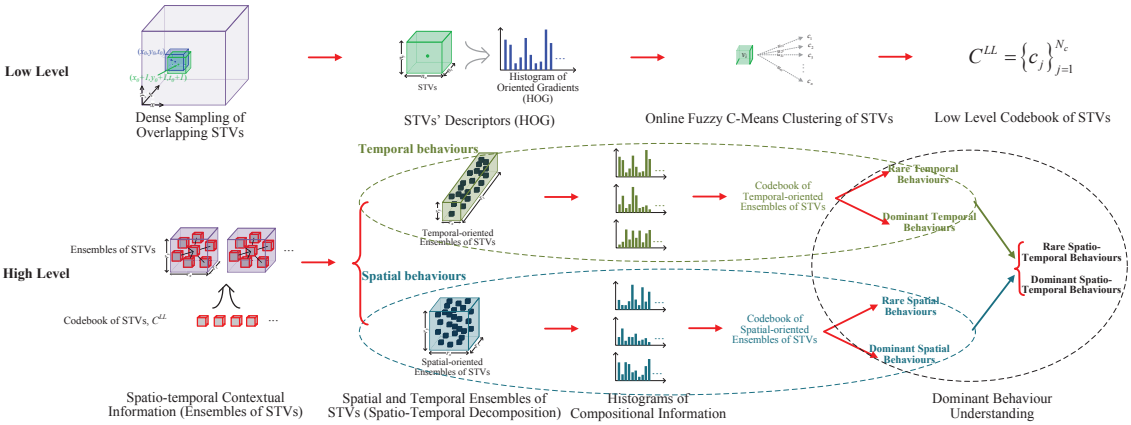


FIGURE 4.2. Algorithm overview: behavior understanding. Behaviors are learnt from local low-level visual information, which is achieved by constructing a hierarchical codebook of the STVs. To capture spatio-temporal configurations of video volumes, a probabilistic framework is employed by estimating probability density functions of the arrangements of video volumes. The uncertainty in the codeword construction of STVs and contextual regions is considered, which makes the final decision more reliable. The high-level output can be employed to simultaneously model normal and abnormal behaviors.

can determine all of the abnormal ("anomalous") spatial and temporal behaviors in a video.

4.3.1 Dynamic Scene Modeling. Considering the structure presented in Figure 4.2, our goal is to the learn dominant behaviors in the scene and from these *determine abnormalities*. We use densely sampled videos and construct a hierarchy of spatio-temporal regions in the video to model dominant local activity patterns. This hierarchical codebook structure has two important characteristics: it codes the compositional information of the video volumes and analyzes the spatial and temporal information independently, thereby making it capable of detecting purely spatial or temporal abnormalities. Moreover, the uncertainty in the codebook construction process is considered in the hierarchical structure. As illustrated, the algorithm for dominant behavior learning consists of two hierarchical levels: low level scene representation, and contextual information of the low level features and clusters of ensemble of volumes.

4.3.1.1 *Low-Level Scene Representation.* The first stage of the algorithm is to represent a surveillance video by meaningful spatio-temporal descriptors. This is achieved by dense sampling, thereby producing STVs, and then clustering similar video volumes. The constructed codebook at this level is called the low-level codebook, $\mathbf{C}^{LL}$, as illustrated in Figure 4.2.

As described in Chapter 2, the 3D STVs, $v_i \in \mathbb{R}^{n_x \times n_y \times n_t}$ are constructed by densely sampling the videos. These volumes are then characterized by the histogram of the spatio-temporal gradient (HOG) of the video in polar coordinates(see section 2.3.1). This descriptor represents both motion and appearance and possesses some degree of robustness to unimportant variations in the data, such as illumination changes [11, 97]. Notwithstanding its simplicity, the results obtained are very promising. However, it should be noted that our algorithm does not rely on a specific descriptor for the video volumes, so that other more complex descriptors might enhance the performance of the approach.

In the previous step, a set of spatio-temporal volumes, v_i, was constructed using dense sampling and represented by a descriptor vector, h_i. Following the paradigm of the "bag of video words" approaches, those video volumes should be grouped based on their similarities. To be capable of handling large amounts of data, and also considering the sequential nature of the video frames, the clustering strategy needs to be capable of limiting the amount of memory used for data storage and computations. Thus, we adopt an online fuzzy clustering approach for very large datasets, which is capable of incrementally updating the cluster centers as new data are observed [39]. The basic idea is to consider a chunk of data, cluster it, and then construct another chunk of data using the new observations. The clusters are then updated [39]. Here we adopt the online single-pass fuzzy clustering algorithm of [40].

Let N_d denote the number of feature vectors in the dth chunk of data and N_C the number of cluster centroids (codewords). These are represented by a set of vectors, $C = \{c_n\}_{n=1}^{N_C}$. We modify the objective function (J) [40] for fuzzy probabilistic

clustering as follows:

$$J = \sum_{i=1}^{N_C} \sum_{j=1}^{N_d} u_{i,j}^m w_j d_{ij} \left(h_j, c_i\right) \tag{4.1}$$

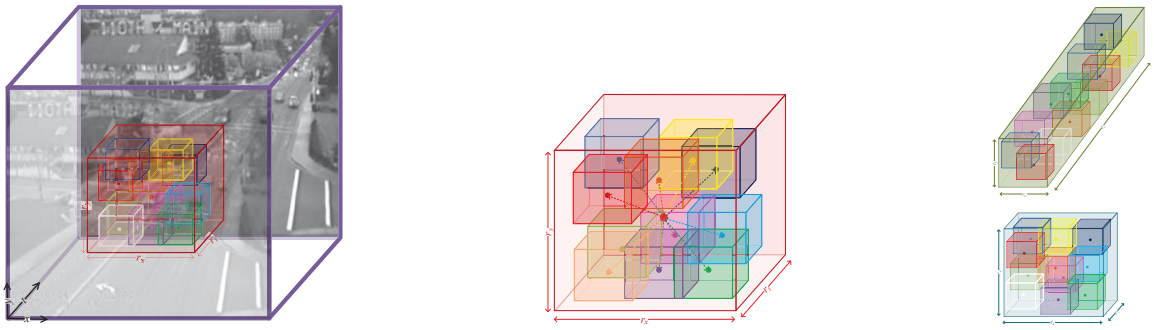
where the parameter w_j is the weight of the jth sample. Note that in the original version, $w_j = 1, \forall j$ [40]. Using the Euclidean distance as the similarity measurement between STVs descriptors, we define the update rule for the cluster center, similarity matrix and the weights w_i as follows:

$$u_{n,j} = \left(\sum_{i=1}^{N_C} \left(\frac{\|h_j - c_n\|}{\|h_j - c_i\|} \right)^{\frac{2}{m-1}} \right)^{-1} \tag{4.2}$$

$$c_n = \frac{\sum_{j=1}^{N_d} w_j u_{n,j}^m h_j}{\sum_{j=1}^{N_d} w_j u_{n,j}^m}, \qquad w_i = \sum_{j=1}^{N_d+N_C} u_{i,j} w_j \tag{4.3}$$

Employing this clustering procedure, a set of clusters is formed for the STVs. These are used to produce is a codebook of STVs and sets of similarity values for every STV. Ultimately, each STV, h_i, will be represented by a set of similarity values: $\{u_{j,i}\}_{j=1}^{N_C}$.

4.3.1.2 *Contextual Information: Ensembles Of Volumes.* As indicated earlier, in order to understand the scene background and make the correct decision regarding normal and suspicious (foreground) events, it is necessary to analyze the spatio-temporal arrangements of volumes [14, 88] in the clusters determined in section 4.3.1.1. The main drawback of many previously reported approaches is that they do not consider the context (spatio-temporal composition of the STVs) at each pixel in the video. Here, we employ the concept of the ensembles of video volumes, as described in section 2.3.2.1. Therefore instead of a single video volume, we consider a large region R around each pixel. R contains many video volumes and thereby captures both local and more distant information in the video frames. Such a set is called an *ensemble* of volumes around the particular pixel in the video (Figure 4.3).



(A) (B) (C)

FIGURE 4.3. Ensembles of video volumes. (A) An ensemble of STVs. (B) Spatio-temporal contextual information. (C) Spatial and temporal oriented ensembles.

The ensemble of volumes ($E_{s,t}$) surrounding each pixel s in the video at time t, is defined as:

$$E_{s,t} = \left\{ v_i^{E_{s,t}} \right\}_{i=1}^{I} \triangleq \{ v_i : v_i \in R_{s,t} \}_{i=1}^{I} \tag{4.4}$$

where $R_{s,t}$ is a region with pre-defined spatial and temporal radii centered at point (s,t) in the video (e.g., $r_x \times r_y \times r_t$), and I indicates the total number of volumes in the ensemble. To capture the spatio-temporal compositions of the video volumes, we use the *relative* spatio-temporal coordinates of the volume in each ensemble [88]. Thus, $x_{v_i}^{E_{s,t}} \in \mathbb{R}^3$ is the relative position of the ith video volume, v_i(in space and time), inside the ensemble of volumes, $E_{s,t}$, for a given point (s,t) in the video (Figure 4.3B). During the codeword assignment process described in the previous section, each volume v_i inside each ensemble was assigned to all labels c_j with weights of $u_{j,i}$ using (4.2). Let the central volume of $E_{s,t}$ be given by v_c. Therefore, the ensemble is characterized by a set of volume position vectors, codewords and their related weights:

$$E_{s,t} = \left\{ x_{v_i^{E_{s,t}}}, u_{ji} \right\}_{i=1:I, j=1:N_C} \tag{4.5}$$

A common approach for calculating similarity between ensembles of volumes is to use the star graph model [14, 75, 11]. This model uses the joint probability between a database and a query ensemble to decouple the similarity of the topologies of the

ensembles and that of the actual video volumes [75]. To avoid such a decomposition, we estimate the *pdf* of the volume composition in an ensemble. Thus, the probability of a particular arrangement of volumes v inside the ensemble of $E_{s,t}$ is given by:

$$P_{E_{s,t}}(v) = P(x_v, c_1, c_2, ..., c_n) = \sum_{i=1}^{n} P(x_v|v = c_i) P(v = c_i) \tag{4.6}$$

The first term in the summation in (4.6), $P(x_v|v = c_i)$, expresses the topology of the ensembles, while the second,$P(v = c_i)$, expresses the similarity of their descriptors (i.e. the weights for the codeword assignments at the first level). We would like to represent each ensemble of volumes by its *pdf*, $P_{E_{s,t}}(v)$. Therefore, given the set of volume positions and their assigned codewords, the probability density function (*pdf*) of each ensemble can be formed using either a parametric model or non-parametric estimation. Here, we approximate the *pdf*s describing each ensemble using (nonparametric) histograms.

4.3.1.3 *Space/Time decomposition of ensembles.* As stated previously, we are interested in detecting *normal* spatial and temporal activities to ultimately distinguish them from both spatial (shape and texture changes) and temporal abnormalities. These are typically foreground regions, and so our approach can also be considered as performing a *focus of attention* task. In order to individually characterize the different behaviors in the video, two sets of ensembles of spatio-temporal volumes are formed, one for the spatially oriented ensembles of volumes and the other, for the temporally oriented ones.

$$\begin{aligned} \mathbf{D}^S &= \{E_{s,t}|r_t \ll min\{r_x, r_y\}\} \\ \mathbf{D}^T &= \{E_{s,t}|r_t \gg max\{r_x, r_y\}\} \end{aligned} \tag{4.7}$$

where $\mathbf{D}^S$ and $\mathbf{D}^T$ represent the sets of spatially- and temporally-oriented ensembles, respectively, and $(r_x \times r_y \times r_t)$ is the size of the ensembles in (4.4). The spatial and temporal decomposition of ensembles of STVs is illustrated in Figure 4.3C.

4.3.1.4 *Clustering ensembles of STVs.* Once a video clip has been processed by the first level of BOV clustering in section 4.3.1.1, each ensemble of spatio-temporal volumes has been represented by a *pdf* of its spatio-temporal volume distribution, as described in 4.3.1.2. Note that such an ensemble *pdf* represents a moving foreground object in the video. The histogram of each ensemble, as obtained from (4.6), is employed as the feature vector to cluster the ensembles. This will then permit us to construct a behavioral model for the video as well as infer the dominant behavior. Using the *pdf* to represent each ensemble of volumes makes it possible to use a divergence function from statistics and information theory as the dissimilarity measure. Here we use the symmetric Kullback-Leibler (KL) divergence to measure the difference between the two *pdf*s [12]. Therefore the distance between two ensembles of volumes, E_{s_i,t_i} and E_{s_j,t_j}, is defined as:

$$d\left(P_{E_{s_i,t_i}}, P_{E_{s_j,t_j}}\right) = KL\left(P_{E_{s_i,t_i}} || P_{E_{s_j,t_j}}\right) + KL\left(P_{E_{s_j,t_j}} || P_{E_{s_i,t_i}}\right) \tag{4.8}$$

where $P_{E_{s_i,t_i}}$ and $P_{E_{s_j,t_j}}$ are the *pdf*s of the ensembles E_{s_i,t_i} and E_{s_j,t_j}, respectively, and d is the symmetric KL divergence between the two *pdf*s in (4.8). The next step is to apply online fuzzy single-pass clustering, as described in section 4.3.1.1, thereby, producing a set of membership values for each pixel. The clustering is performed independently for the two sets of ensembles, $\mathbf{D}^S$ and $\mathbf{D}^T$, obtained from (4.7). The resulting two codebooks are then represented by $\mathbf{C}^S = \left\{c_{k_S}^S\right\}_{k_S=1}^{N_S}$ and $\mathbf{C}^T = \left\{c_{k_T}^T\right\}_{k_T=1}^{N_T}$, respectively.

4.3.2 Behavior Analysis. The result of the processing in section 4.3.1 permits us to construct a set of behavior patterns for each pixel. As stated previously, we are interested in detecting *dominant* spatial and temporal activities as an ultimate means of determining both spatial (shape and texture changes) and temporal abnormalities (foreground regions). Next, we consider the scenario of a continuously operating surveillance system. At each temporal sample t, a single image is added to the already observed frames and a new video sequence, the *query*, Q, is formed. The

query is densely sampled in order to construct the video volumes and thereby, the ensembles of STVs, as described in section 4.3.1.

Given the already existing codebooks of ensembles constructed in 4.3.1.4, each pixel in the query, q_i is characterized by a set of similarity matrices, $\mathbf{U}_{q_i}^S = \left\{u_{k_S,i}^S\right\}_{k_S=1}^{N_S}$ and $\mathbf{U}_{q_i}^T = \left\{u_{k_T,i}^T\right\}_{k_T=1}^{N_T}$. We note that $u_{k_S,i}^S$ and $u_{k_T,i}^T$, respectively, are the similarity of the observation to the k_S spatial and k_T temporal cluster of ensembles. Then the description that best describes a new observation is given by:

$$(k_S^*, k_T^*) = \arg\left(\max_{k_S}\left\{u_{k_S,i}^S\right\}, \max_{k_T}\left\{u_{k_T,i}^T\right\}\right) \tag{4.9}$$

To infer normality or abnormality of the query, q_i, two similarity thresholds, Θ_{k_S} and Θ_{k_T}, are employed:

$$\left(\alpha u_{k_S^*,i}^S + \beta u_{k_T^*,i}^T\right) \underset{rare}{\overset{dominant}{\gtrless}} \left(\alpha\Theta_{k_T^*} + \beta\Theta_{k_S^*}\right) \tag{4.10}$$

where α and β are preselected weights for the spatial and temporal codebooks, respectively and Θ_{k_S} and Θ_{k_T} are the *learnt* likelihood thresholds for the kth codeword of the spatial and temporal codebooks, respectively. To determine these, we employ the set of previously observed pixels, $\mathbf{D} = \{p_i\}$, as represented by the two cluster similarity matrices obtained in section 4.3.1.4, $\mathbf{U}_{p_i}^S = \left\{u_{k_S,i}^S\right\}_{k_S=1}^{N_S}$ and $\mathbf{U}_{p_i}^T = \left\{u_{k_T,i}^T\right\}_{k_T=1}^{N_T}$. Thus, the previous observations can be divided into N_S and N_T disjoint subsets:

$$\begin{aligned}
\mathbf{D}_{k_S} &= \left\{p_i | u_{k_S,i}^S > \varepsilon\right\}_{p_i\in\mathbf{D}}, \bigcup_{k_S=1}^{N_S} \mathbf{D}_{k_S} = \mathbf{D} \\
\mathbf{D}_{k_T} &= \left\{p_i | u_{k_T,i}^T > \varepsilon\right\}_{p_i\in\mathbf{D}}, \bigcup_{k_T=1}^{N_T} \mathbf{D}_{k_T} = \mathbf{D}
\end{aligned} \tag{4.11}$$

where $\mathbf{D}_{k_S}$ and $\mathbf{D}_{k_T}$ contain only the most representative examples of each cluster, k_S and k_T respectively. Clearly, representativeness is governed by the parameter ε.

Then, similar to [72], we construct the likelihood thresholds as follows:

$$\Theta_{k_S} = \frac{\gamma}{|\mathbf{D}_{k_S}|} \sum_{i \in \mathbf{D}_{k_S}} \log u^S_{k_S,i} + \frac{1-\gamma}{|\mathbf{D}| - |\mathbf{D}_{k_S}|} \sum_{i \notin \mathbf{D}_{k_S}} \log u^S_{k_S,i}$$
$$\Theta_{k_T} = \frac{\gamma}{|\mathbf{D}_{k_T}|} \sum_{i \in \mathbf{D}_{k_T}} \log u^T_{k_T,i} + \frac{1-\gamma}{|\mathbf{D}| - |\mathbf{D}_{k_T}|} \sum_{i \notin \mathbf{D}_{k_T}} \log u^T_{k_T,i} \tag{4.12}$$

where the parameter $\gamma \in [0, 1]$ controls the abnormality/normality detection rate and $|\mathbf{D}|$ indicates the number of members of $\mathbf{D}$. Returning to (4.10), the parameters α and β are seen to control the balance between spatial and temporal abnormalities based on the ultimate objective of the abnormality detection. As an example, if the objective is to detect the temporal abnormality in the scene (background/foreground segmentation), then one can assume that $\alpha = 0$.

4.3.3 Online Model Updating. In this section we describe how the algorithm is updated in an online manner. The scenario we have considered implies on-line and continuous surveillance of a particular scene in order to simultaneously detect dominant and anomalous patterns. As described in section 4.3.1, the algorithm only requires the first N frames of the video stream to initiate the process. This is achieved by constructing the codebook of STVs (section 4.3.1.1), ensembles of volumes (section 4.3.1.2) and finally the codebook of ensembles (section 4.3.1.3).

When new data are observed, the past N_d frames are always employed to update the learnt codebooks, i.e. the clusters of both STVs and ensembles of STVs. This process is performed continuously and the detection thresholds, Θ_{k_S} and Θ_{k_T} are updated in an ongoing manner as described in (4.12) based on the previously learnt codebooks.

(A) (B) (C)

FIGURE 4.4. Dominant behavior understanding on data captured by a camera during different times of the day. The lighting conditions change gradually from daylight to night. (A) A sample frame. (B) The dominant behaviors are produced by the cars passing through the lanes running from top to bottom and vise versa. (C) The abnormalities are those cars entering the intersection from the left.

4.4 Experiments

The algorithm has been tested using the same dataset as of section 3.4: the dominant behavior understanding dataset in [129][3], UCSD pedestrian dataset [65][4], and subway surveillance videos [2][5]. In all cases, we have assumed that local video volumes are of size $5 \times 5 \times 5$ and the HOG is calculated assuming $n_\theta = 16$, $n_\phi = 8$ and $N_d = 50$ frames. Parameters α and β were selected depending on the desired goal of the abnormality detection. These were set empirically to 0.1 and 0.9 for motion detection and to 0.5 for abnormal activity detection. Quantitative evaluation and comparison of different approaches are presented in terms of precision-recall and ROC curves, obtained by varying the parameter γ in (4.12)[6].

The first dataset consists of three videos sequences. The first one, *Belleview*, is a traffic scene in which lighting conditions gradually change during different times of the day. The dominant behaviors are either the static background or the dynamic cars passing through the lanes running from top to bottom. Thus, the rare events

[3] http://www.cse.yorku.ca/vision/research/spatiotemporal-anomalous-behavior.shtml

[4] http://www.svcl.ucsd.edu/projects/anomaly

[5] Obtained from the authors of [2]

[6] To make a quantitative comparison possible, the algorithm is evaluated for abnormality detection and compared to the state-of-the-art.

("abnormalities") are the cars entering the intersection from the left. Figure 4.4 (a), (b), and (c) illustrate a sample frame, and the dominant and abnormal behavior maps, respectively. In the *Boat-Sea* video sequence, the dominant behavior is the waves while the abnormalities are the passing boats since they are newly observed objects in the scene. The *Train* sequence, is one of the most challenging videos available [129] due to drastically varying illumination and camera jitter. The background changes rapidly as the train passes through tunnels. In this sequence the abnormality relates to people movement. Figure 4.5 shows a sample video frame of each video sequence, the detected abnormal regions and the precision/recall curves. We followed the same initialization strategy as [129] and compared the results with two alternative pixel-level anomaly detection methods: spatio-temporal oriented energies in [129] and local optical flow in [2]. As the abnormalities in this dataset are low level motions, we also include the pixel-level background models (Gaussians Mixture Models [134]) and the behavior template approaches in [49] for comparison.

Comparing the performance of the different approaches in Figure 4.5C, we observe that, in general, our method was comparable or superior to the others shown. In particular, the method based on spatio-temporal oriented energy filters [129] produced results comparable to ours, but might not be useful for more complex behaviors for two reasons: it is too local and does not consider contextual information.

It is also clear that conventional methods for background subtraction (GMM) fail to detect dominant behaviors in scenes containing complicated behaviors, such as the *Train* and *Belleview* video sequences. However, they still do produce good results for background subtraction in a scene with a stationary background (*Boat-Sea* video sequences). In the latter case, the so-called abnormality (the appearance of the boat)is sufficiently different from the scene model. Thus, GMM seems promising for this video. On the other hand, we observe that simple local optical flow features, combined with online learning [2], do not yield acceptable results in the scenes with dynamic backgrounds. It appears that the optical flow approach has difficulty capturing temporal flicker and dynamic textures.

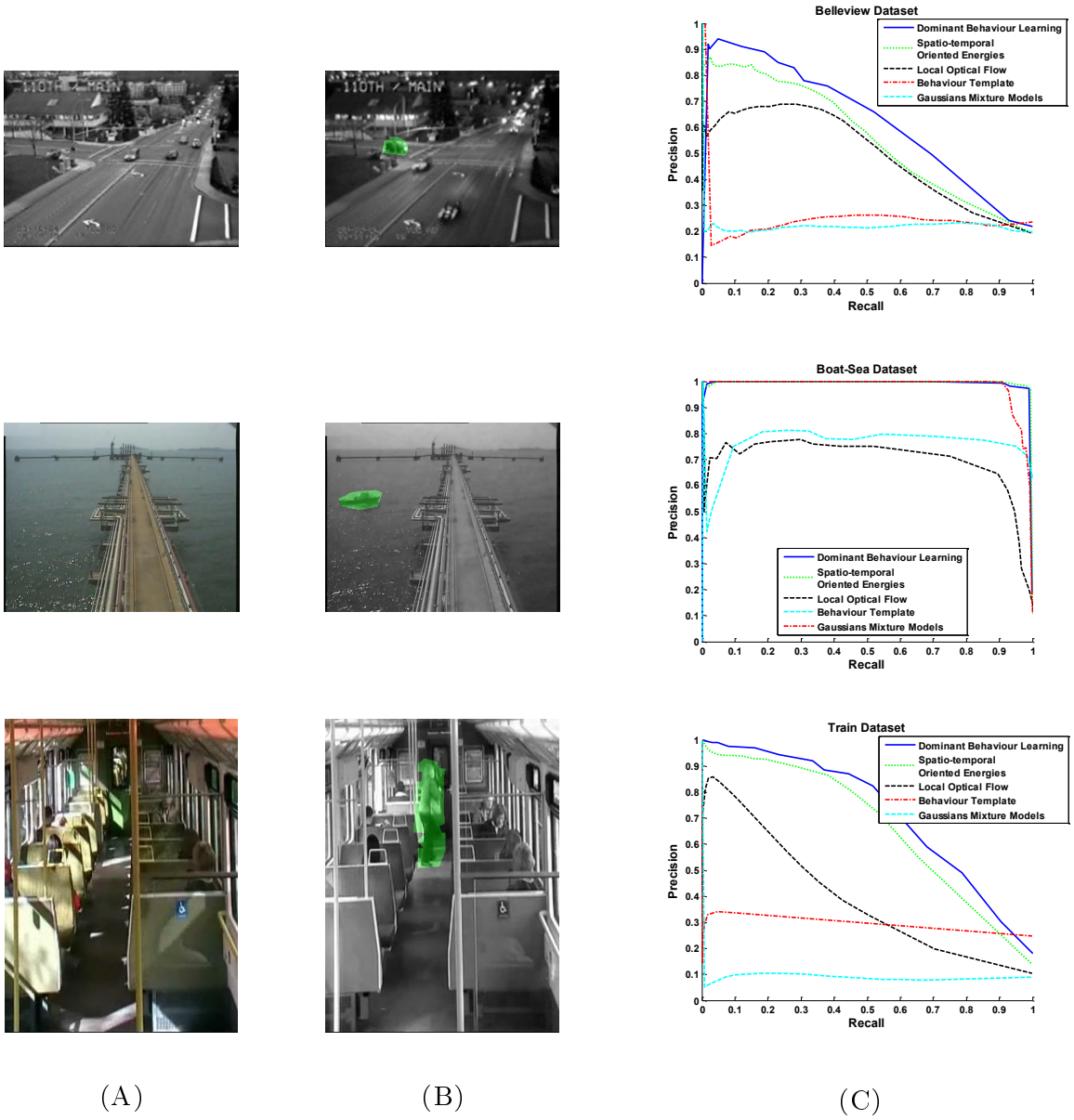


FIGURE 4.5. Dominant behavior understanding and abnormality detection. Experiments with three videos are illustrated from top to bottom in the figure: *Belleview*, *Boat-Sea* and *Train*. The first experiment (first row) is concerned with detecting dominant and abnormal behavior in a busy traffic scene. The second and third experiments were conducted on videos in which the abnormalities were defined as being rare but nevertheless acceptable foreground motions. The anomalous regions are highlighted in green. Column (A) Sample frames from the three videos. Column (B) The detected anomalous regions are cars moving from right to left (top), a boat moving to the right (middle), and a moving person (bottom). Column (C) Precision/recall curves.

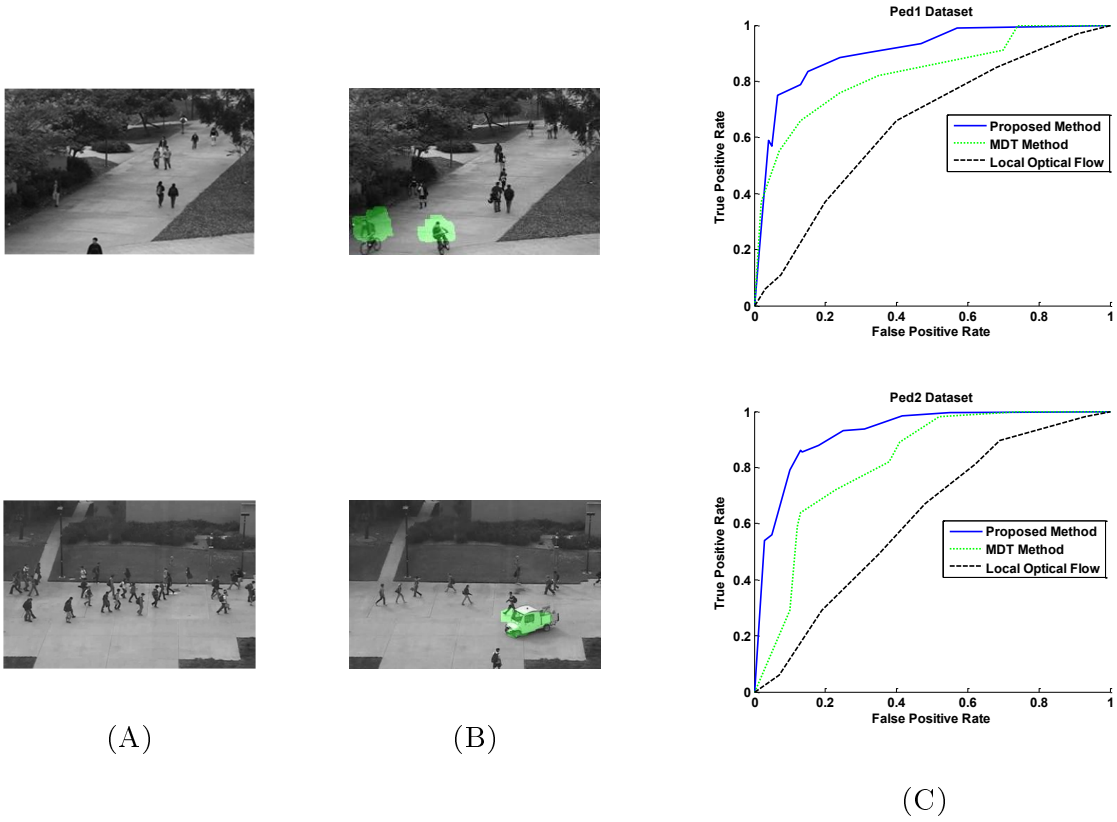


FIGURE 4.6. Frame level abnormality detection using the UCSD pedestrian datasets. Top: Ped1 dataset, Bottom: Ped2 dataset. (A) Sample frames. (B) Detected anomalous regions: bicyclist (top), a car (bottom). (C) ROC curves for the proposed approach and alternatives (MDT [65], Local optical flow [2]).

We also conducted experiments with the UCSD pedestrian dataset[7]. It contains video sequences from two pedestrian walkways where abnormal events occur. The dataset exhibits different crowd densities, and the anomalous patterns are the presence of non-pedestrians on a walkway (bikers, skaters, small carts, and people in wheelchairs). Figure 4.6 contains samples of two videos with the detected suspicious regions as well as the ROC curves for different methods (Figure 4.6C). In order to

[7]This dataset was employed as it includes pixel level ground truth showing the exact location of the abnormal regions in each frame.

TABLE 4.1. Quantitative comparison of the proposed method and the state-of-the-art for anomaly detection using the Ped1 dataset. (* indicates that the method is claimed to have real time performance).

Algorithm	EER (frame-level)	EER (pixel-level)
*Proposed algorithm	15%	29%
MDT (Mahadevan *et al.*, 2010, [65])	25%	58%
Sparse Reconstruction (Cong *et al.* 2011 [24])	19%	-
*Bertini *et al.*, 2012, [11]	31%	70%
*Reddy *et al.*, 2011, [85]	22.5%	32%
ST-MRF (Kim and Grauman, 2009, [52])	40%	82%
*Local optical flow, (Adam *et al.* 2008 [2])	38%	76%
Saligrama and Chen, 2012, [94]	16%	-

make a quantitative comparison the equal error rate (EER) was also calculated for both pixel and frame level detection as suggested by [65][8].

The results in Table 4.1 indicate that the proposed algorithm outperformed all other *real-time algorithms* and achieved the best results for the UCSD pedestrian dataset at both frame level detection and pixel level localization. Furthermore, the number of *initialization frames* required by the proposed algorithm is significantly lower than the alternatives (200 frames compared to 6400 frames). This is a major advantage of the proposed method that can also learn dominant and abnormal behaviors on the fly. Moreover the computational time required by the method described in this chapter is significantly lower than others in the literature. In summary, our experiments signify that our approach is capable of reasonably handling drastically and gradually changing backgrounds and illumination conditions, as well as detecting abnormal events with different spatial and temporal complexities, ranging from the scene background to human activities. Furthermore, the algorithm is adaptive. It does not require a long training video and updates itself after observing a small number of initialization frames.

[8]Frame level detection implies that a frame is marked as suspicious if it contains any abnormal pixel, regardless of its location. On the other hand, pixel level detection attempts to measure the localization ability of an algorithm. This requires that the detected pixels in each video frame be compared to a pixel level ground truth map.

4.5 Summary

This chapter presents a novel approach for simultaneously learning dominant behaviors and detecting anomalous patterns in videos. This algorithm is centered on three main ideas: hierarchical analysis of multi-scalar visual features; accounting for their spatio-temporal compositional information of the low level features; and spatial and temporal decomposition of the behaviors in order to learn dominant spatial and temporal activities. First, spatio temporal video volumes are constructed for densely sampled videos and then, dominant behaviors are learnt based on a hierarchical analysis of spatio-temporal video volumes and their compositional information. By employing different analyses in the spatial and temporal domains, the algorithm is capable of learning different behaviors and detecting pure spatial and temporal abnormalities. This hierarchical property makes the algorithm extendible, which means higher levels of analysis can be performed with the results. A major advantage of the algorithm is that it can simultaneously model *high-level behaviors* and *detect* abnormalities by considering both spatial and temporal contextual information, while also performing *temporal pixel level change detection* and background subtraction. This characteristic makes our algorithm more general than both abnormality detection and background subtraction methods on their own. A limitation of the current approach is that it does not account for trajectories and hence, long term behaviors are not learnt. Future research will extend the approach by adding another level of analysis in the hierarchical structure to model the spatial and temporal connectivity of the learnt behaviors.

We have tested the algorithm on four popular benchmarks and shown that the algorithm is both effective and robust for both anomaly detection and localization tasks. Moreover, the results are highly competitive with state-of-the-art methods. However, a major advantage of our approach is that it does not require any feature analysis, background/foreground segmentation and tracking, and is susceptible to on-line real-time analysis.

Chapter 5

Video To Video Matching And Activity Recognition

5.1 Introduction

Given the tremendous number of potential practical video applications, there is a great demand for automated systems that analyze and understand the contents of these videos. Obviously, recognizing and localizing human actions in a video are of primary importance to such a system. Although there exist many methods for accomplishing this in highly controlled environments, this task still remains a challenge in real world environments, which are subject to camera motion, cluttered backgrounds, occlusion, and scale/ viewpoint/ perspective variations [74, 95]. Moreover, the same action performed by two different persons can appear to be very different, and clothing, illumination and background can substantially increase this dissimilarity and make the problem extremely difficult [14, 99].

In this chapter, our main goal is to address the problem of *action recognition* and *localization* in real environments using a hierarchical probabilistic video-to-video matching framework. This problem is also referred to as *action spotting* [26]. To achieve this, we have developed a fast data-driven approach, which finds similar videos in a "target" set to a single labelled "query" video. Assuming that the latter contains an action of interest, e.g., walking, we find all videos in the target set that

that are similar to the query, which implies the same activity. This video-to-video comparison also makes it possible to *label* activities, the so-called action classification problem. An overview of the algorithm is presented in Figure 5.1. The major benefit of our approach is that it does not require long video training sequences, object segmentation, tracking or background subtraction. The method can be considered as an extension to the original *Bag of Video Words* approach for action recognition.

Although an initial spatio-temporal volumetric representation of human activity may eliminate some pre-processing steps, for example background subtraction and tracking, it suffers from some major drawbacks. For example, in general, BOW-based approaches for activity recognition in the literature involve salient point detection. They usually ignore the geometrical and temporal structure of these visual volumes, as they store STVs in an unordered manner. Also they are unable to handle scale variations (spatial, temporal, or spatio-temporal) because they are too local, in the sense that they consider just a few neighboring video volumes (e.g., five nearest neighbors in [75] or just one neighbor in [95]). To overcome these issues, we have developed a multi-scale, hierarchical codebook of BOWs for *densely sampled* videos, which incorporates spatio-temporal *compositions* and their *uncertainties*. This permits the use of statistical inference to recognize the activities. We also note that, in order to measure similarity between a query and a target dataset, it is necessary to use information regarding the most *informative* spatio-temporal video volumes (STVs) in the video, i.e., the salient foreground objects. To select these space-time regions, we use the information obtained from our hierarchical BOV method, which in a sense, can be viewed as being a context-based spatio-temporal segmentation method.

As shown in Figure 5.1, the proposed algorithm consists of two main components, hierarchical codebook construction of salient STVs and an inference mechanism for measuring the similarity between salient STVs of the query and target videos. Hierarchical codebook construction consists of four steps: coding the video to construct STVs and low-level probabilistic codebook formation while considering the uncertainties in the STVs; constructing ensembles of video volumes for each pixel in a video

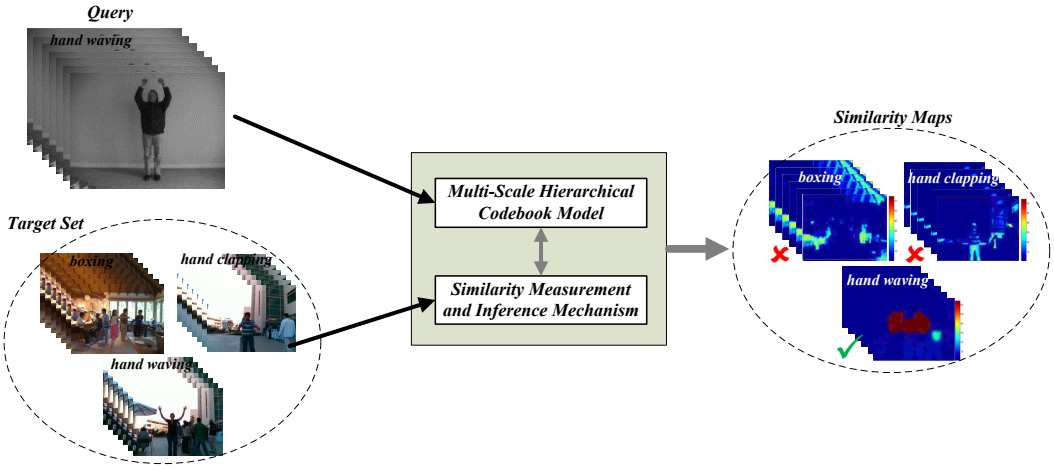


FIGURE 5.1. Overview. The goal is to find similar videos to the query video in the target set. This is achieved by constructing an activity model for the query video and then measuring the similarity between it and the target videos.

frame containing a large number of STVs and probabilistic models of their spatio-temporal compositions; high-level codebook construction of the ensembles; and finally, analyzing codewords as a function of time in order to construct a codebook of salient regions. The inference mechanism is based on a set of hierarchical codewords constructed for each query video. It determines the most similar compositions of STVs in the target videos that match the query video. There are two important differences between our proposed hierarchical approach and previously reported ones. First, the latter are unable to handle both local and global compositional information. Second, they always select the informative regions at the lowest level of the hierarchy.

The main characteristics of this algorithm are as follows:

- We introduce a hierarchical codebook structure for action detection and labelling. This is achieved by considering a large volume containing many STVs and constructing a probabilistic model of this volume to capture the spatio-temporal configurations of STVs. Consequently, similarity between two videos is calculated by measuring the similarity both between spatio-temporal video volumes and their compositional structures.

- We select the salient pixels in the video frames by analyzing codewords obtained at the highest level of the hierarchical codebook's structure. This differs from conventional background subtraction and salient point detection methods.

In order to evaluate the capability of our approach for action matching and classification we have conducted experiments using three datasets: KTH [96], Weizmann [36] and MSR II [127][1]. Three types of experiments were performed: action matching and retrieval, single dataset video classification, and cross-dataset action recognition.

5.2 Related Work

Many studies have focused on the action recognition problem by invoking human body models, tracking-based methods, and local descriptors [80]. The early work often depended on tracking [82, 83, 125, 110], in which humans, body parts, or some interest points were tracked between consecutive frames to obtain the overall appearance and motion trajectory. Clearly, the performance of these algorithms is highly dependent on tracking, which sometimes fails for real world video data [119]. Recently, tracking a fixed number of interest point between video frames has become more popular than other tracking-based approaches since they are capable of coding some contextual information regarding local spatio-temporal features. This method functions by tracking the interest point features between consecutive frames and thereby obtaining a set of trajectories [68, 104, 110]. The contextual information is then computed as the spatial relationship between trajectories [68] or temporal associations between interest points on a single trajectory [104]. In addition to the normal issues associated with tracking, these approaches are based on an implicit assumption of a static background, since moving objects in the background might produce trajectories similar to an object in the region of interest.

Alternatively, shape template matching has been employed for activity recognition which have been described in section 2.1. Although it seems that they are likely

[1] http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/

well suited to action localization, they do require a priori high-level representations of the human motion. Moreover, they depend on such image pre-processing stages as segmentation, object tracking, and background subtraction [120], which are extremely challenging in real-world unconstrained environments.

In order to eliminate such pre-processing, Derpanis et. al. [26] have proposed so-called "action templates". These are calculated as oriented local spatio-temporal energy features that are computed as the response of a set of tuned 3D Gaussian third order derivative filters. Sadanand et. al. [93] introduced action banks to make these template-based recognition approaches more robust to viewpoint and scale variations. Recently, tracking and template-based approaches have been combined to improve the action detection accuracy [125, 51].

As indicated in section 2.2, models based on a bag of local visual features have recently been studied extensively and shown promising results for action recognition [14, 16, 21, 50, 54, 74, 75, 95, 99, 120, 123]. These approaches extract and quantize the video data to produce a set of video volumes that form a "visual vocabulary". In general, the potential *real-time* performance of these methods is related to the number of video volume samples and their associated features [50]. Usually, these features are gradients (spatial, temporal, or spatio-temporal), body landmarks, or color information. Combining them makes it possible to capture motion and the scene context simultaneously without requiring reliable trajectories of the objects of interest [38]. The video volumes are constructed either by extracting a limited set of *interest points* or densely sampling the video. In the former, due to the sparse nature of the space-time interest points, the method becomes computationally efficient and hence is popular in the action recognition literature [74, 96, 112, 123, 126]. On the other hand, the selection of appropriate interest points that are *guaranteed* to contain a salient and discriminative motion pattern in their local context is a difficult challenge [56]. In addition, it has been shown recently that densely sampling the video always achieves better results than a sparse set of interest points [111].

As described previously, a major advantage of using volumetric representations of videos is that it permits the localization and classification of actions using data-driven nonparametric approaches instead of requiring the training of sophisticated parametric models. However, we note that classical BOV approaches suffer from a significant challenge. That is, the video volumes are grouped solely based on their similarity, in order to reduce the vocabulary size. To overcome this challenge, contextual information must be included in the original BOV framework. One solution is to employ visual phrases instead of visual words. This has proposed in [130] where a visual phrase is defined as a set of spatio-temporal video volumes with a specific pre-ordained spatial and temporal structure. The main drawback of this approach is that it cannot localize different activities in a video frame. Alternatively, the solution presented by Boiman and Irani [14] is to densely sample the video and store *all* video volumes for a video frame, along with their relative locations in *space* and *time*. However, the main problem with this approach is that it requires excessive computational time and a considerable amount of memory to store all of the volumes as well as their spatio-temporal relationships. We present a competent alternative to this in the next section.

In addition to [14], several other methods have been proposed to incorporate spatio-temporal structure in the context of BOV. These are often based on co-occurrence matrices that are employed to describe contextual information. For example, the well-known correlogram exploits spatio-temporal co-occurrence patterns [95]. In [34] the spatial information is coded through the concatenation of video words detected in different spatial regions as well as data mining techniques, which are used to find frequently occurring combinations of features. Similarly, [66] addresses this issue by using the spatial configuration of the 2D patches by incorporating their weighted sum. In the same way, in [33] coding of spatial information is achieved through the concatenation of video words detected in different spatial regions. Data mining is used to find frequently occurring combinations of features. In [56], these patches were represented using 3D Gaussian distributions of the spatio-temporal gradient and

the temporal relationship between these Gaussian distributions was modeled using HMMs. An interesting alternative is to incorporate mutual contextual information of objects and human body parts by using a random tree structure [120, 123] to partition the input space. The likelihood of each spatio-temporal region in the video is then calculated. The primary issue with this approach [123] is that it requires background subtraction, interest point tracking and detection of regions of interest.

Hierarchical clustering seems to be an attractive way of incorporating the contextual structure of video volumes, as well as preserving their compactness of their description [54, 75]. Thus a modified version of [14] was presented in [75]. It uses a hierarchical approach, in which a two-level clustering method is employed. At the first level, all similar volumes are categorized. Then clustering is performed on randomly selected groups of spatio-temporal volumes while considering the relationships in space and time between the five nearest spatio-temporal volumes. However, the small number of spatio-temporal volumes involved again makes this method local in nature. Another hierarchical approach is presented in [54], which attempts to capture the compositional information of a subset of the most discriminative video volumes. In all of these proposed solutions to date, although a higher level of quantization in the action space produces a compact subset of video volumes, it also significantly reduces the discriminative power of the descriptors, an issue addressed in [15]. Generally, all of the earlier work described above for modeling the mutual relationships between the video volumes have one or more limitations such as: considering relationships between only a pair of local video volumes [62, 95]; being too local and unable to capture interactions of different body parts [54, 32]; and considering either spatial or temporal order of volumes [95].

In this chapter we present a hierarchical probabilistic codebook method for action recognition and localization in videos. The proposed codebook structure has two important characteristics: it codes the compositional information of the 3D video volumes and selects the most informative ones in the video.

5.3 Multi Scale Hierarchical Codebooks

Considering the structure presented in Figure 5.1, our aim is to find the similarity between the query and all of the target videos. Our work is based on the bag of space-time features approach in that a set of STVs is used for measuring similarity. The proposed recognition algorithm in Figure 5.1 consists of two main steps: densely sampling videos from which hierarchical codebooks are constructed (see Figure 5.2) and using an inference mechanism for finding the appropriate action in the target videos. Although the main idea of the former part is introduced in Chapter 2, in this section, we briefly describe hierarchical codebook structure and section 5.4 describes the inference mechanism.

5.3.1 Low-Level Scene Representation. The first stage of the algorithm is to represent a query video by meaningful spatio-temporal descriptors. This is achieved by dense sampling, thereby producing a large number of spatio-temporal video volumes. Then similar video volumes are clustered to from a codebook. Since this is actually done on-line, frame-by-frame, the codebook is adaptive. The constructed codebook at this level is called the low-level codebook, as illustrated in Figure 5.2. This is achieved by following the procedure of codebook construction in section2.3.1.

Similar to all BOV approaches, 3D STVs in a video are constructed at the lowest level of the hierarchy. The descriptor vector for each video volume, taken as a histogram of oriented gradients (HOG), is constructed using (2.2).

Following the procedure explained in section 3.3.1.1, similar video volumes are grouped to construct a codebook. The codebook is continuously being pruned to eliminate codewords that are either infrequent or very similar to the others, which ultimately generates $M^{\mathcal{L}}$ different codewords that are taken as the labels for the video volumes, $\mathbf{C}^{\mathcal{L}} = \{c_i\}_{i=1}^{M^{\mathcal{L}}}$.

After the initial codebook formation, [2], each new 3D volume, v_i, can be assigned to all labels, c_j's, with a degree of similarity, $w_{i,j}$, as shown in Figure 2.2A. We note

[2]Recall that initialization requires a minimum of one video frame.

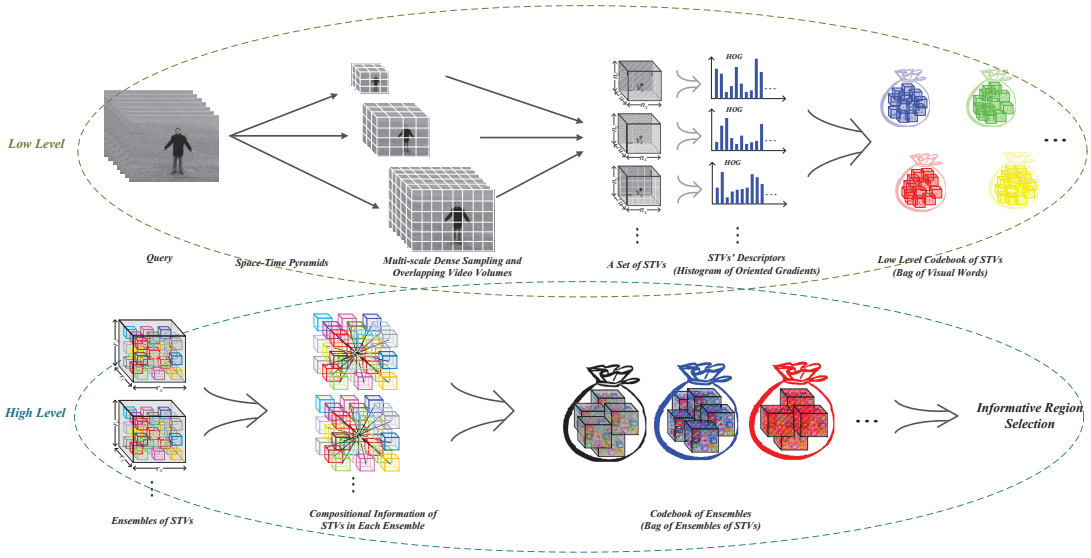


FIGURE 5.2. Overview of the scene representation and hierarchical codebook structure. First, the query video is densely sampled at different spatio-temporal scales followed by the construction of a set of overlapping spatio-temporal video volumes. Subsequently, a two level hierarchical probabilistic codebook is created for the video volumes. At the lower level of the hierarchy, similar video volumes are grouped to form a conventional low level codebook, $\mathbf{C}^{\mathcal{L}}$, but while considering the uncertainty in codeword assignment. At the higher level, a much larger spatio-temporal 3D volume around each pixel, containing many STVs, is considered in order to capture the spatio-temporal arrangement of the volumes. We refer to this graph as an ensemble of volumes. Using these graphs, similar ensembles are grouped based on the similarity between arrangements of their video volumes and yet another codebook is formed. The most informative codewords are then selected by examining the temporal correspondence between codewords. Note: This is a copy of the Figure 2.1 for reader's convenience.

that the number of labels (shown in color), $M^{\mathcal{L}}$, is much less than the number of volumes, N.

5.3.2 High-Level Scene Representations. At the previous step, similar video volumes were grouped in order to construct the low level codebook. The outcome of this is a set of similar volumes, clustered regardless of their positions in space and time. This is the point at which all other BOV methods stop. As stated in Chapter 2, the main drawback of many BOV approaches is that they do not consider the spatio-temporal composition (context) of the video volumes. Certain methods for

capturing such information have appeared in the literature (see [14, 58, 66]). Here we employ the probabilistic framework introduced in section 2.3.2 for quantifying the arrangement of the spatio-temporal volumes.

Suppose a new video is to be analyzed; we refer to it as the *query*. The goal is to measure the likelihood of each pixel in the target videos given the query. To accomplish this, it is necessary to analyze the spatio-temporal arrangement of the volumes in the clusters that have been determined in section 5.3.1. Thus, we next consider a large 3D volume around each pixel in (x, y, t) space. This large region contains many volumes with different spatial and temporal sizes as shown in Figure 2.2B. Thus it captures both the local and more distant information in the video frames. Such a set is called an *ensemble* of volumes around the particular pixel in the video and defined by (2.3).

To capture the spatio-temporal compositions of the video volumes, we use the *relative* spatio-temporal coordinates of the volume in each ensemble, as shown in Figure 2.2C. Therefore, each ensemble of video volumes at point (x_i, y_i, t_i) is represented by a set of such video volumes and their relative positions, and hence (2.3) can be rewritten as:

$$E(x_i, y_i, t_i) = \left\{ \Delta_{v_j}^{E_i}, v_j, v_o \right\}_{j=1}^{J} \tag{5.1}$$

An ensemble of volumes, $E\left(x_i, y_i, t_i\right)$ is characterized by a set of video volumes, the central video volume, v_o, and the relative distance of each of the volumes in the ensemble, v_j, to the central video volume, $\Delta_{v_j}^{E_i} \in \mathbb{R}^3$, as represented in (5.1). This provides a view-based graphical spatio-temporal multi-scale description at each pixel in every frame of a video. Following the procedure of section 2.3.2.2, each ensemble of volumes can be represented by a set of *pdf*s as follows:

$$P\left(\mathbf{\Gamma}|E_i\right) = \bigcup_{\substack{m=1:M^{\mathcal{L}} \\ n=1:M^{\mathcal{L}}}} \left\{ P\left(\Gamma_{m,n}\left(\Delta\right)|E_i\right) \right\} \tag{5.2}$$

where $P\left(\mathbf{\Gamma}|E_i\right)$ is s set of *pdf*s modeling topology of the ensemble of volumes. Here, the probabilistic vote for a spatio-temporal position (the first factor on the right hand

side of (2.14)), $P\left(\Delta|c_m, c_n, \Delta_{v_j}^{E_i}\right)$, is approximated using (nonparametric) histograms. Given the representation of an ensemble of volumes in (5.2), similarity between two video sequences can be computed simply by matching the *pdf*s of the ensembles of volumes at each pixel.

Once a video clip has been processed, each ensemble of spatio-temporal volumes has been represented by a set of *pdf*s as given in (5.2). Having performed the first level of clustering in section 5.3.1, and given the representation of each ensemble obtained in (5.2), the aim now is to cluster the ensembles. This will then permit us to construct a behavioral model for the query video. Although clustering can be performed using many different approaches [117, 71], spectral clustering methods are currently in vogue due to their superior performance to traditional methods. Moreover, they can be computed efficiently. Spectral clustering constructs a similarity matrix of feature vectors and seeks an optimal partition of the graph representing the similarity matrix using eigen-decomposition [109]. Usually, this is followed by either k-means or fuzzy c-means clustering. We utilize the normalized decomposition method of [73].

Employing the overall *pdf* $P(\mathbf{\Gamma}|E_i)$ in (5.2) to represent each ensemble of volumes makes it possible to use divergence functions from statistics and information theory as the appropriate dissimilarity measure. Here we use the symmetric Kullback-Leibler (KL) divergence to measure the difference between the two *pdf*s, f and g [12]:

$$d(f,g) = KL(f||g) + KL(g||f) \tag{5.3}$$

where $KL(f||g)$ is the Kullback-Leibler (KL) divergence of f and g. Therefore, given the *pdf* of each ensemble of volumes in (5.2) the similarity between two ensembles of volumes, $E(x_i, y_i, t_i)$ and $E(x_j, y_j, t_j)$, is defined as:

$$s_{E_i,E_j} = e^{-\frac{d^2\left(P(\mathbf{\Gamma}|E_i),P\left(\mathbf{\Gamma}|E_j\right)\right)}{2\sigma^2}} \tag{5.4}$$

where $P\left(\mathbf{\Gamma}|E\left(x_i, y_i, t_i\right)\right)$ and $P\left(\mathbf{\Gamma}|E\left(x_j, y_j, t_j\right)\right)$ are the *pdf*s of the ensembles $E\left(x_i, y_i, t_i\right)$ and $E\left(x_j, y_j, t_j\right)$, respectively, obtained in section 5.3.2. d is the symmetric KL divergence between the two *pdf*s in (5.3) and σ is the variance of the KL divergence over all of the observed ensembles of STVs in the query.

Given the similarity measurement of the ensembles in (5.4), the similarity matrix, $\mathbf{S}_N$, for a set of ensembles of volumes is formed and the Laplacian calculated as follows:

$$L = D^{-\frac{1}{2}}\mathbf{S}_N D^{\frac{1}{2}} \tag{5.5}$$

where D is a diagonal matrix whose ith diagonal element is the sum of all elements in the ith row of $\mathbf{S}_N$. Subsequently, an eigenvalue decomposition is applied to L and the eigenvectors corresponding to the largest eigenvalues are normalized and form a new representation of the data to be clustered [73]. This is followed by online fuzzy single-pass clustering [40] to produce $M^{\mathcal{H}}$ different codewords for the high-level codebook of ensembles of STVs, $\mathbf{C}^{\mathcal{H}} = \{c_i\}_{i=1}^{M^{\mathcal{H}}}$, for each pixel.

5.3.3 Informative Codeword Selection. In order to select a particular video in a target set that contains a similar activity to the one in the query video, the uninformative regions (e.g., background) must obviously be excluded from the matching procedure. This is conventionally performed in all activity recognition algorithms. Generally, for shape-template and tracking based approaches this is done at the pre-processing stages using such methods as background subtraction and ROI selection. These have their inherent problems discussed in section 5.2. On the other hand, selecting informative rather than uninformative regions is a normal aspect of almost every BOV-based approach that constructs STVs at interest points. Clearly, these are intrinsically related to the most informative regions in the video. When we consider the framework for activity recognition proposed in this chapter, the high-level codebook of ensembles of STVs is used to generate codes *for all pixels in each video frame*. Therefore it is crucial to select only the most informative codewords and their related pixels. Given the high-level codebook, $\mathbf{C}^{\mathcal{H}}$, constructed in section 5.3.2, we saw that a codeword is assigned to each pixel $p(x, y)$ at time (t) in the

video. Therefore, in a video sequence of temporal length $\mathcal{T}$, a particular pixel $p(x, y)$ is represented as a sequence of assigned codewords at different times:

$$p\,(x, y) = \left\{p(x, y) \leftarrow c_i : \forall t \in \mathcal{T},\ c_i \in \mathbf{C}^{\mathcal{H}}\right\} \tag{5.6}$$

A sample video frame and the assigned codewords are illustrated In Figure 5.3. In order to remove non-informative codewords (e.g., codewords which represent the scene background), each pixel and its assigned codewords are analyzed as a function of time. As an example, Figure 5.3 plots the assigned codewords to the sampled pixels in the video over time. It is observed that the pixels related to the background or static objects show stationary behavior. Therefore the associated codewords can be removed by employing a simple temporal filter at each pixel. This method was inspired by the pixel-based background model presented in [53], where a time series of each of the three quantized color features was created at each pixel. A more compact model of the background is then determined by temporal filtering, based on the idea of the *Maximum Negative Run-Length* (MNRL). The MNRL is defined as the maximum amount of time between observing two samples of a specific codeword at a particular pixel [53]. The larger the MNRL, the more likely the codeword is not the background. The main difference from [53] is that we employ the assigned codewords as the representative features for every pixel, as obtained from the high level codebook $\mathbf{C}^{\mathcal{H}}$ (see (5.6)).

The major advantage of selecting informative codewords at the highest level of the coding hierarchy is that compositional scene information comes into play[3]. Hence the computational cost is greatly reduced and the need for a separate background subtraction algorithm is eliminated.

In summary, at first, the query video is densely sampled at different spatio-temporal scales in order to construct the video volumes. Then a low level codebook is formed and each volume v_j is assigned to a codeword c_i, $c_i \in \mathbf{C}^{\mathcal{L}}$, with similarity

[3]Some advanced approaches for background modeling also incorporate spatio-temporal compositions of the motion-informative regions to build a background model [70, 91].

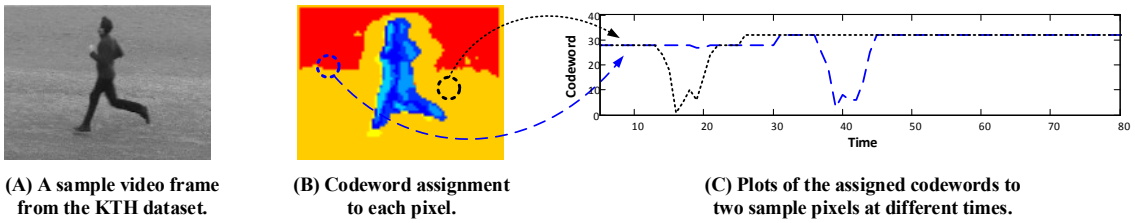


FIGURE 5.3. Informative codeword selection. (A) A sample video frame from KTH dataset in which the person is running. (B) High-level codewords assigned to every pixel in the video frame. (C) Temporal correspondence of the codewords at each pixel. A time series of the assigned codewords from the high level codebook is ascribed to each pixel in the video. Pixels related to the background or static objects show a stationary behavior over time, and hence, they are assumed to be uninformative.

$w_{j,i}$. Then a larger 3D volume around each pixel, containing many STVs, the so-called ensemble of STVs, is considered. The spatio-temporal arrangement of the volumes inside each ensemble is model based on a set of *pdf*s. At the next level of the hierarchical structure, another codebook is formed for these ensembles of STVs, $\mathbf{C}^{\mathcal{H}}$. The two codebooks are then employed for finding similar videos to the query.

Two main features characterize the constructed probabilistic model of the ensembles. First the spatio-temporal probability distribution is defined independently for each codebook entry. Second, the probability distribution for each codebook entry is estimated using (non-parametric) histograms. The former renders the approach capable of handling certain deformations of an object's parts while the latter makes it possible to model the true distribution instead of making an oversimplifying Gaussian assumption.

5.4 Similarity map construction and video matching

The overall goal is to find similar videos to a query video in a target set and consequently label them according to the labelled query video using the hierarchical codebook presented in section 5.3. Figure 5.4 summarizes the process of determining the hierarchical codebooks and how the similarity maps are constructed.

Hierarchical codebook construction for the query video

For a query video, $\mathcal{Q}$, containing a particular action,

- Densely sample the video at all scales and construct spatio-temporal video volumes and their descriptors: $\mathcal{Q} = \{v_j\}_{j=1}^{N_\mathcal{Q}}$
- Construct the low level codebook of video volumes for the query: $\mathbf{C}^\mathcal{L}$
- Construct ensembles of spatio-temporal volumes: $E(x, y, t)$
- Construct the topological models of the ensembles of volumes as described in section 3.2.2.
- Construct the high level codebook, $\mathbf{C}^\mathcal{H}$, to cluster similar ensembles of volumes.
- Remove non-informative codewords from $\mathbf{C}^\mathcal{H}$.

This procedure results in two codewords for a query video containing a particular activity: $\{\mathbf{C}^\mathcal{L}, \mathbf{C}^\mathcal{H}\}$

Similarity map construction for a target video

For each video, $\mathcal{V}$, in the target dataset,

- Densely sample the video at all scales and construct spatio-temporal volumes: $\mathcal{V} = \{v_j\}_{j=1}^{N_\mathcal{V}}$
- Assign each video volume in the target video to the low level codewords of each subset of query videos: $v_j \leftarrow c_k, \quad c_k \in \mathbf{C}^\mathcal{L}$
- Construct an ensemble of volumes at each particular pixel, $E(x, y, t)$
- Construct the topological models of the ensembles of volumes
- Measure similarity between ensembles of STVs in the target video to the high level codebook and assign the most similar codeword to the ensemble:
  $E(x, y, t) \leftarrow c_{k^*}$
  $k^* = \arg\max_k s_{E(x,y,t),c_k}, \quad c_k \in \mathbf{C}^\mathcal{H}$
- The similarity map between the query and target at each point is then constructed as: $\mathcal{S}_{\mathcal{Q},\mathcal{V}}(x, y, t) = s_{E(x,y,t),c_{k^*}}$

FIGURE 5.4. The complete algorithm for similarity measurement between query and target videos. The query video is densely sampled and two codebooks are formed. The similarity between a target video and query at each pixel is measured based on these and then employed to construct a similarity map.

The inference mechanism is the procedure for calculating similarity between particular spatio-temporal volume arrangements in the query and the target videos. More precisely, given a query video containing a particular activity, $\mathcal{Q}$, we are interested in constructing a dense *similarity map* for every pixel in the target video, $\mathcal{V}$, by utilizing *pdf*s of the volume arrangements in the video. At first, the query video is densely sampled and a low level codebook is constructed for local spatio-temporal video volumes. Then the ensemble of video volumes is formed. These data are used to create a high level codebook, $\mathbf{C}^\mathcal{H}$, for coding spatio-temporal compositional information of the video volumes, as described in section 5.3. Finally, the query video is represented

by its associated codebooks[4]. In order to construct the similarity map for the target video, $\mathcal{V}$, it is densely sampled at different spatio-temporal scales and the codewords from $\mathbf{C}^{\mathcal{L}}$ are assigned to the video volumes. Then the ensembles of video volumes are formed at every pixel and the similarity between the ensembles in $\mathcal{V}$ and the codewords in $\mathbf{C}^{\mathcal{H}}$ is measured using (5.4). In this way, a similarity map is constructed at every pixel in the target video, $\mathcal{S}_{\mathcal{Q},\mathcal{V}}(x,y,t)$. The procedure for similarity map construction has been described in detail in Figure 5.4. Note again that no background and foreground segmentation and no explicit motion estimation are required in the proposed method.

Having constructed a similarity map, it remains to find the best match to the query video[5]. Generally two scenarios are considered in activity recognition and video matching: (1) Detecting and localizing an activity of interest and (2) Classifying a target video given more than one query, which is usually referred to as action classification. For both of these, the region in the target video that contains a similar activity to the query must be selected at an appropriate scale. We perform multi-scale activity localization, so that ensembles of volumes are generated at each scale independently. Hence, we produce a set of independent similarity maps for each scale. Therefore, for a given ensemble of volumes, $E(x,y,t)$ in the target video, a likelihood function is formed at each scale:

$$p\left(\mathcal{S}_{\mathcal{Q},\mathcal{V}}(x,y,t) \mid scale\right) \tag{5.7}$$

where $\mathcal{S}_{\mathcal{Q},\mathcal{V}}(x,y,t)$ is the similarity between the ensemble of volumes in the target video, $E(x,y,t)$, and the most similar codeword in the high-level codebook, $c_{k^*} \in \mathbf{C}^{\mathcal{H}}$, and *scale* represents the scale at which the similarity is measured. In order to localize the activity of interest, i.e., finding the most similar ensemble of volumes in the target video to the query, the maximum likelihood estimate of the scale at each pixel

[4]The query is represented by two codebooks: the low level codebook of spatio-temporal video volumes, $\mathbf{C}^{\mathcal{L}}$, and the high level codebook of the ensembles of video volumes, $\mathbf{C}^{\mathcal{H}}$.

[5]The inference mechanism is relatively simple as our aim is to introduce and formulate a hierarchical structure for constructing a similarity map between videos based on densely sampled STVs and their spatio-temporal compositions. However, it could be replaced by a more sophisticated one.

is employed. Therefore, the most appropriate scale at each pixel is the one that maximizes the following likelihood estimate:

$$scale^* = \arg\max_{scale} p\left(\mathcal{S}_{\mathcal{Q},\mathcal{V}}\left(x,y,t\right) \mid scale\right) \tag{5.8}$$

In order to find the most similar ensemble to the query, a detection threshold was employed. Hence, an ensemble of volumes is said to be similar to the query and contains the activity of interest if $\mathcal{S}_{\mathcal{Q},\mathcal{V}}\left(x,y,t\right) \geq \gamma$ at $scale^*$. In this way, the region in the target video that matches the query is detected[6].

For action classification problem, we consider a set of queries, $\mathbf{Q} = \bigcup\{\mathcal{Q}_i\}$, each containing a particular activity[7]. Then the target video is labeled according to the most similar video in the query. For each query video, $\mathcal{Q}_i$, two codebooks are formed and then the similarity maps are constructed as described in Figure 5.4. This produces a set of similarity maps for all activities of interest. Therefore, the target video contains a particular activity, i^*, that maximizes the accumulated similarity between all ensembles of volumes in the target video as follows:

$$i^* = \arg\max_{i} \left(\sum_{E(x,y,t)\in\mathcal{V}} \mathcal{S}_{\mathcal{Q}_i,\mathcal{V}}\left(x,y,t\right) \right), \quad \mathcal{Q}_i \in \mathbf{Q} \tag{5.9}$$

Despite the simple inference mechanism employed here for action recognition and localization, the obtained experimental results show the strength of our approach for similarity map construction between two videos. We also note that the proposed statistical model of codeword assignment and the arrangement of the spatio-temporal volumes permit small local misalignments in the relative geometric arrangement of the composition. This property, in addition to the multi-scale volume construction in each ensemble, enables the algorithm to handle certain non-rigid deformations in space and time. This, of course, is necessary since human actions are not exactly reproducible, even for the same person. Obviously, activity recognition from a single example

[6]The threshold, γ was set empirically to 0.7 of the maximum similarity value for every query video in all experiments.

[7]In most of the reported approaches for activity recognition it is implicitly assumed that the query contains a single activity.

eliminates the need for a large number of training videos for model construction and significantly reduces computational costs. On the other hand, it imposes some limitations by its nature. It appears that learning from a single example is not as general as the models constructed using many training examples, and therefore our approach may not be as general as the model-based approaches. However, it should be emphasized that constructing a generic viewpoint and scale invariant model for an activity requires a large amount of labeled training data, which do not currently exist. Moreover, imposing strong priors by assuming particular types of activities reduces the search space of possible poses considered, which limits their application to action recognition.

We conclude this section by examining the computational complexity of our algorithm. Suppose there are K video volumes available in each ensemble and the number of codewords in the low- and high- level codebooks are $M^{\mathcal{L}}$ and $M^{\mathcal{H}}$, respectively. For each ensemble, the time complexity of the low level and high level codeword assignment are $O\left(K \times M^{\mathcal{L}}\right)$, and $O\left(M^{\mathcal{H}}\right)$, respectively. Therefore the complexity of calculating each point in a similarity map is $O\left(K \times M^{\mathcal{L}} \times M^{\mathcal{H}}\right)$.

5.5 Experimental Results

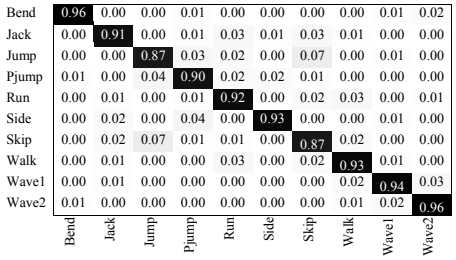
The algorithm was tested on three different datasets: KTH [96], Weizmann [36] and MSR II [127] to determine its capabilities for action recognition. The Weizmann and KTH datasets are the standard benchmarks in the literature used for action recognition. The Weizmann dataset consists of ten different actions performed by nine actors, and the KTH action data set contains six different actions, performed by twenty-five different persons in four different scenarios (indoor, outdoor, outdoor at different scales, outdoor with different clothes). The MSR II consists of 54 video sequences, recorded in different environments with cluttered backgrounds in crowded scenes, and contains three types of actions similar to the KTH: boxing, hand clapping, and hand waving. We evaluated our approach for three different scenarios. The first one is "action matching and retrieval using a single example", in which both target and

query videos are selected from the same dataset. This task measures the capability of the proposed approach for video matching. The second scenario is the "single dataset action classification" task in which more than one query video is employed to construct the model of a specific activity. Here, single dataset classification implies that both query and target videos are selected from the same dataset. Finally, in order to measure the generalization capability of our algorithm to find similar activities in videos recorded in different environments, "cross-dataset action detection" was performed. This scenario implies that that the query and target videos could be selected from different datasets.

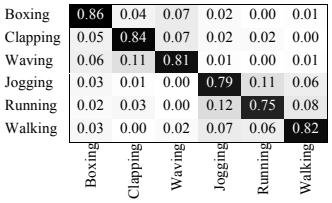
Video matching and classification were performed using KTH and Weizmann, which are single-person, single-activity videos. We used them to compare with the current state-of-the-art even though they were collected in controlled environments. For cross-dataset action recognition, we used the KTH dataset as the query set, while the target videos were selected from the more challenging MSR II dataset. Our experiments demonstrate the effectiveness of our hierarchical codebook method for action recognition in these various categories. In all cases, we have assumed that local video volumes are of size $n_x = n_y = n_t = 5$, and the HOG is calculated assuming $n_\theta = 16$, $n_\phi = 8$. The ensemble size was set to $r_x = r_y = r_t = 50$. The number of codewords in the low- and high-level codebooks were set to 55 and 120, respectively[8]. Later in this section we will thoroughly examine the effect of different parameters on the performance of the algorithm.

5.5.1 Action Matching And Retrieval Using A Single Example. Since our proposed method is a video-to-video matching framework, it is not necessary to have a training sequence. This means that we can select one labelled query video for each action, and find the most similar one to it in order to perform the labelling. For the Weizmann dataset, we used one person for each action as a query video and the rest (eight other persons) as the target sets. This was done for all persons in the dataset and the results were averaged. The confusion matrix for the Weizmann

[8]These parameters are similar to the ones in a similar study [88]



(A) Weizmann dataset



(B) KTH dataset

FIGURE 5.5. Confusion matrices for single video action matching, (A) Weizmann dataset, (B) KTH dataset. A single video is used as a query to which the other videos in the dataset were matched.

TABLE 5.1. Action recognition comparison with the state-of-the-art for single video action matching (percentage of the average recognition rate).

Method	Dataset	
	KTH	Weizmann
Proposed method	81.2	91.9
Thi et.al. [105]	77.17	88.6
Seo et.al. [99]	69	78

dataset is shown in Figure 5.5A, achieving an average recognition rate of 91.9% over all 10 actions. The columns of the confusion matrix represent the instances to be classified, while each row indicates the corresponding classification results.

We carried out the same experiment on the KTH dataset. The confusion matrix is shown in Figure 5.5B. The average recognition rate was 81.2% over all 6 actions. The results indicate that the method proposed in this chapter outperforms state-of-the-art approaches, even though the former requires no background/foreground segmentation and tracking. The average accuracy of the other methods is presented in Table 5.1.

The overall results on the Weizmann dataset are better than those on the KTH dataset. This is predictable, since the Weizmann dataset contains videos with more static backgrounds and more stable and discriminative actions than the KTH dataset.

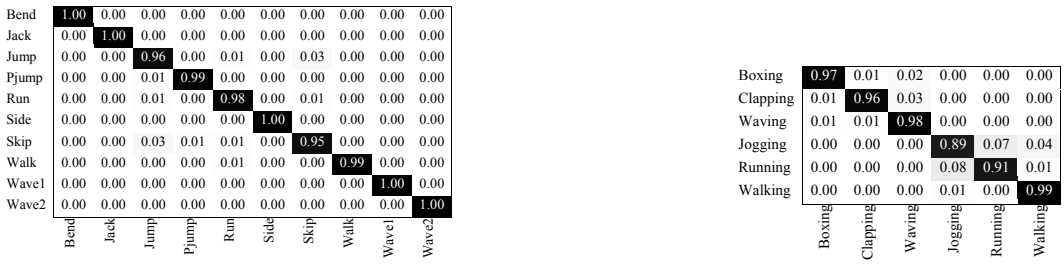
TABLE 5.2. Single video action matching in the KTH dataset when target videos are limited to four subsets, each obtained under different recording conditions. The query video is selected from one of the four subsets of videos with a different recording condition. Then the most similar video from each target is found and used as the label applied to the query (percentage of the average recognition rate).

		Target			
		s1	s2	s3	s4
Query	s1	88.5	71.4	82.1	83.6
	s2	72.1	74.2	69.7	71.6
	s3	81.9	70.5	77.1	80.6
	s4	82.3	73.6	81.1	84.8

In order to measure the capabilities of our approach in dealing with scale and illumination variations, we reported the average recognition rate for different recording scenarios in the KTH dataset. According to [96], KTH contains four different recording conditions are: s1) outdoors; s2) outdoors with scale variations; s3) outdoors with different clothes; and s4) indoors. The evaluation procedure employed here is to construct four sets of target videos, each having been obtained under the same recording condition. Then, a query is selected from one of these four scenarios and the most similar video to the query is found in each target dataset in order to perform the labelling. The average recognition rates are reported in Table 5.2. When the target and query videos are selected from the same subset of videos with the same recording conditions, the average recognition rate is higher than when they are taken under different recording conditions. Moreover, although we have claimed that our method is scale- and illumination-invariant, it appears that, in these experiments, the recognition rate decreases when the query and target videos have been taken under different recording conditions. This is particularly evident when the target videos are recorded at different scales (see the second column in Table 5.2). Thus scale and clothing variations degrade the performance of our algorithm more than changes in illumination. Therefore, as we might have expected, an activity model constructed using just a *single example* cannot adequately account for all scale/illumination variations in a scene.

5.5.2 Single Dataset Action Classification. In order to make an additional quantitative comparison of our algorithm with the state-of-the-art, we have extended it to the action classification problem. This refers to the more classical situation in which we use a *set of query videos* instead of just a single one, as discussed previously. We have evaluated our algorithm's ability to apply the correct label to a given video sequence, when both the training[9] and target datasets are obtained from the same dataset. We tested the Weizmann and KTH datasets, and applied the standard experimental procedures in the literature. For the Weizmann dataset, the common approach for classification is to use leave-one-out cross-validation, i.e., eight persons are used for training and the videos of the remaining person are matched to one of the ten possible action labels. Consistent with other methods in the literature, we mixed the four scenarios for each action in the KTH dataset. We followed the standard experimental procedure for this dataset [96], in which 16 persons are used for training and nine for testing. This is done 100 times and after which the average performance over these random splits is calculated [96]. The confusion matrix for the Weizmann dataset is reported in Figure 5.6A and the average recognition rate is 98.7% over all 10 actions in the leave-one-out setting. As expected from earlier experiments reported in the literature, our results indicate that the "skip" and "jump" actions are easily confused, as they appear visually similar. For the KTH dataset, we achieved an average recognition rate of 95% for the six actions as shown in the confusion matrix in Figure 5.6. As observed from Figure 5.6B, the primary confusion occurs between jogging and running, which was also problematical for the other approaches. Obviously, this is due to the inherent similarity between the two actions. The recognition rate was also compared to other approaches (see Table 5.3). Comparing our results with those of the state-of-the-art, we observe that they are similar, though again we do not require any background/foreground segmentation and tracking.

[9] Although our method does not actually require any specific training sequences, we refer to the query videos as the training set for consistency with the literature.



(A) Weizmann dataset (B) KTH dataset

FIGURE 5.6. Confusion matrices for action classification, (A) Weizmann dataset, (B) KTH dataset.

TABLE 5.3. Comparison of action recognition with the state-of-the-art (percentage of the average recognition rate). For the KTH dataset, the evaluation is made using either *leave-one-out* or *data-split* as described in the original paper [96].

Method	Evaluation approach	Dataset	
		KTH	Weizmann
Proposed method	split	95.0	98.7
Seo et. al. [99]	split	95.1	97.5
Thi et. al. [105]	split	94.67	98.9
Tian et. al. [106]	split	94.5	-
Liu et. al. [62]	leave one out	94.2	-
Zhang et. al. [130]	split	94.0	-
Wang et. al. [112]	split	93.8	-
Yao et. al. [120]	split	93.5	97.8
Bregonzio et. al. [16]	leave one out	93.17	96.6
Ryoo et. al. [92]	split	91.1	-
Yu et. al. [124]	leave one out	95.67	-
Mikolajczyk et. al. [69]	split	95.3	-
Jiang et. al. [46]	leave one out	95.77	-

5.5.3 Cross-Dataset Action Matching And Retrieval. Similar to other approaches for action recognition [106], we use cross-dataset recognition to measure the robustness and generalization capabilities of our algorithm. In this paradigm, the query videos are selected from one dataset (the KTH dataset in our experiments) and the targets from another (MSR II dataset), so that we compare similar actions performed by different persons in different environments. We selected three classes of actions from the KTH dataset as the query videos: boxing, hand waving, and hand

TABLE 5.4. Percentage of the average correct recognition rate for cross dataset action recognition over three different activities. The query and the target videos are selected from the KTH and MSR II datasets, respectively.

Method	Accuracy (%)
Proposed method	79.8
Tian et al. [106]	78.8
Yuan et al. [126]	59.6

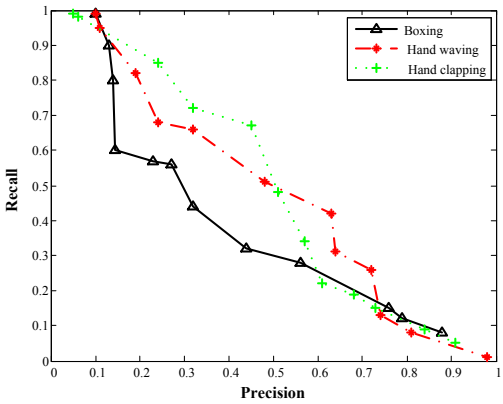


FIGURE 5.7. The precision-recall curves for cross-dataset action recognition. The query videos are selected from the KTH dataset and the targets from the MSR II dataset. Three activities were selected for classification: boxing, hand waving, and hand clapping,

clapping, including 25 persons performing each action. A hierarchical codebook was created for each action category and the query was matched to the target videos. We varied the detection threshold, γ, to obtain the precision/recall curves for each action type, as shown in Figure 5.7. This achieved an overall recognition rate of 79.8%, which is comparable to the state-of-the-art (see Table 5.4).

5.5.4 Effect Of Parameter Variation. As our proposed method creates two codebooks to group similar video volumes and ensembles of video volumes, it is necessary to analyze the effect of different codebook sizes on the performance of the algorithm. Therefore, the overall recognition rate for different codebook sizes was determined as described previously using the KTH dataset,. Various codebook sizes ($M^{\mathcal{H}}$ and $M^{\mathcal{L}}$) were employed and the average recognition rate calculated. In Figure 5.8, the average recognition rate is plotted as a function of the both low-

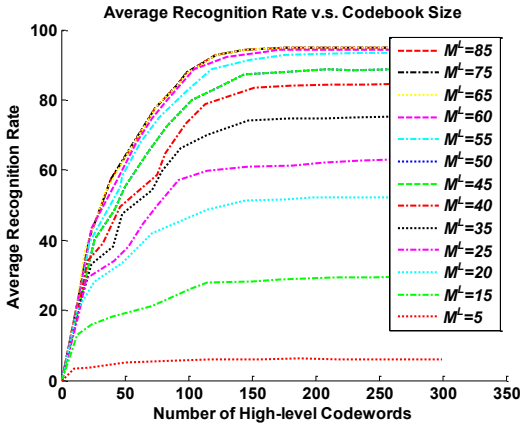


FIGURE 5.8. Effect of different codebook sizes for both low- and high-level codebooks. The average recognition rate is calculated for different codebook sizes for KTH dataset.

and high-level codebook sizes (number of codewords). We observe that small low level codebooks will not produce acceptable results, even with a large number of high level codewords. Therefore preserving information at the lowest level is necessary to achieve acceptable results. Recall that we have shown in the previous section how the number of codewords affects the computational cost of our algorithm.

Similarly, using larger high level codebooks demands more memory and dramatically increases computational time. Therefore the number of codewords must be kept as small as possible. Although there is a trade-off between codeword size and the performance of the algorithm, it can be inferred from our experiments that using relatively small codebooks at both low and high levels, (e.g., $M^{\mathcal{L}} = 55$ and $M^{\mathcal{H}} = 120$) achieves acceptable results for action recognition.

5.6 Summary

In this chapter we have presented a new hierarchical approach based on spatio-temporal volumes for the challenging problem of video-to-video matching and tested for the problem of human action recognition in videos. At the lowest level in the data hierarchy, our approach is an extension of conventional BOW approaches. However, this is only at the bottom level of a more descriptive data hierarchy that is based

on representing a video by compositional contextual data. The hierarchical structure consists of three main levels:

- Densely sampling and coding a video using spatio-temporal volumes to produce a low-level codebook. This codebook is similar to the one constructed in conventional BOW approaches.
- Constructing an ensemble of video volumes and representing their structure using probabilistic modeling of the compositions of the spatio-temporal volumes. This is followed by the construction of a high-level codebook for the volume ensembles.
- Analyzing the codewords assigned to each pixel as a function of time in order to determine salient regions.

Given a single query video (an example of a particular activity), the method computes the similarity of each pixel in each frame of the target videos to the query, and finds the subset of target videos that are similar to that query. This is accomplished by analyzing a relatively large contextual region around the pixel, while considering the compositional structure using a probabilistic framework. The algorithm was tested on three popular benchmarks, KTH, Weizmann, and MSR II. We showed that it is effective and robust for both action-matching and cross-dataset recognition. Moreover, the results are highly competitive with state-of-the-art methods. However, a major advantage of our approach is that it does not require background and foreground segmentation and tracking, and is susceptible to on-line real-time analysis. The proposed video method can easily be extended to multi-action retrieval and action localization by modifying the inference mechanism. Since the proposed method codes the video using spatio-temporal video volumes and their compositional information, it does not impose any constraints on the video contents and therefore, it can be extended to unconstrained video matching and content-based search engines. One of the major advantages of the proposed algorithm for event recognition in videos is that it does not require a model of the event. However, it does have some drawbacks that need to be addressed in future work. Clearly, such a video representation of activities in

a scene cannot be applied for long-term behavior understanding, e.g., behaviors that consist of numbers of activities that occur sequentially. Some form of event segmentation might deal with this issue. Future research will extend the approach by adding another level of analysis to the hierarchical structure, which models the spatial and temporal connectivity of the learnt activities.

Chapter 6

Multi-Object Tracking

6.1 Introduction

Object tracking is, perhaps, the most fundamental task for any high-level video content analysis system. Generally speaking, the use of visual tracking is pertinent in long-term tasks such as activity recognition, automated surveillance systems and abnormality detection, sport analysis and content-based video retrieval. It has been massively studied in the last three decades and a diverse set of approaches and a rich collection of tracking algorithms have been produced. Visual tracking, in general, is a very challenging problem due to the loss of information caused by the projection of the 3D world onto a 2D image, noise in images, cluttered backgrounds, complex object motion, partial or full occlusions, changes in illumination, real-time processing requirements, etc. In the early years, almost all visual tracking methods assumed that the object could be easily discriminated from the background and then recognized. As a result, these approaches were limited to scenes with relatively few constituents, simple motion patterns and smooth object appearance changes. However, tremendous progress has been made in recent years. For example, some algorithms can deal with abrupt appearance changes, object disappearance from scenes, and drifting.

Tracking is a more or less solved problem when objects in a scene are isolated and easily distinguishable from the background. However, in complex and crowded scenes of people there are many objects with similar appearance that can occlude each other.

In addition, occlusions can also be the result of static objects in the scene. Therefore, multiple object tracking remains a challenging problem in computer vision [22].

Decades of research on this topic have produced a diverse set of approaches and a rich collection of tracking algorithms. Readers can refer to [121] and [119] for a review of the state-of-the-art in object tracking and a detailed analysis and comparison of various representative methods.

In the majority of the traditional approaches, only the object itself and/or its background are modeled. Thus we observe that significant progress has been made in this case. For example, many research articles have addressed face, human body, head, and rigid object tracking, which can be categorized within a paradigm of detect-then-track. This is usually done by construing a tracker based on a pre-trained detection and recognition mechanism for the objects of interest, based on appearance modeling of the target [45, 64, 102]. This class of tracking methods are referred to as "object-centric" approaches [57].

On the other hand, detection cannot be performed when there is no prior knowledge about the specific objects being tracked. These methods are referred to as "generic object tracking" or "model-free tracking". Since manually annotating sufficient numbers of examples of all objects in the world is prohibitively expensive and impractical, recently, approaches for model-free tracking have received increased interest [57, 63]. Model-free tracking is a challenging task because there is a little information available about the object to be tracked [63]. Another challenge in multiple-target model-free tracking is the presence of an unknown and ever changing number of targets.

Here we concentrate on creating long-term trajectories for unknown moving objects by using a model-free tracking algorithm. As opposed to the tracking-by-detection algorithms [47, 118], no object detection is involved. Each individual object is tracked only by modeling the temporal relationship between sequentially occurring local motion patterns. This is achieved by constructing two sets of initial tracks that

code local and global motion patterns in videos. These local motion patterns are obtained by analyzing spatially and temporally varying structures in videos. Initially, the video is densely sampled, spatio-temporal video volumes (STVs) are constructed, and similar ones are grouped to reduce the dimension of the search space. This is called the low-level codebook. Then, a large contextual region containing many STVs (in space and time) around each pixel is examined and their compositional relationships are approximated using a probabilistic framework. They are then employed to form yet another codebook, called the high-level codebook. Therefore, two codewords are assigned to each pixel, one from the low level and the other from the high level codebook. By examining pairs of sequential video frames, the matching codewords for each video pixel are transitively linked into distinct tracks, whose total number is unknown a priori and which we will refer to as linklets. The linking process is separately performed for both codebooks. This is done under the hard constraint that no two linklets may share the same pixel at the same time, i.e. the assigned codewords. The end result at this step is two sets of independent linklets obtained from the low- and high-level codebooks. Subsequently, a set of sparse tracks, referred to as tracklets in the literature, are produced by grouping the linklets that indicate similar motion patterns (see Figure 6.1). This produces two sets of independent tracklets, referred to as low- and high-level tracklets. We adopt Markov Chain Monte Carlo Data Association (MCMCDA) to estimate an initially unspecified number of trajectories. To this end, we formulate the tracklet association problem as a Maximum A Posteriori (MAP) problem to produce a chain of tracklets. The final output of the data association algorithm is a partition of the set of tracklets such that those belonging to each individual object have been grouped together.

The main contribution is an approach capable of learning long-term trajectories of any moving object in a video without using any prior knowledge about the objects (object detection). This is achieved by creating local trajectories of regions that have similar motion patterns, while also considering their neighboring regions (contextual information). Therefore, this algorithm is a complete *bottom up* tracking method

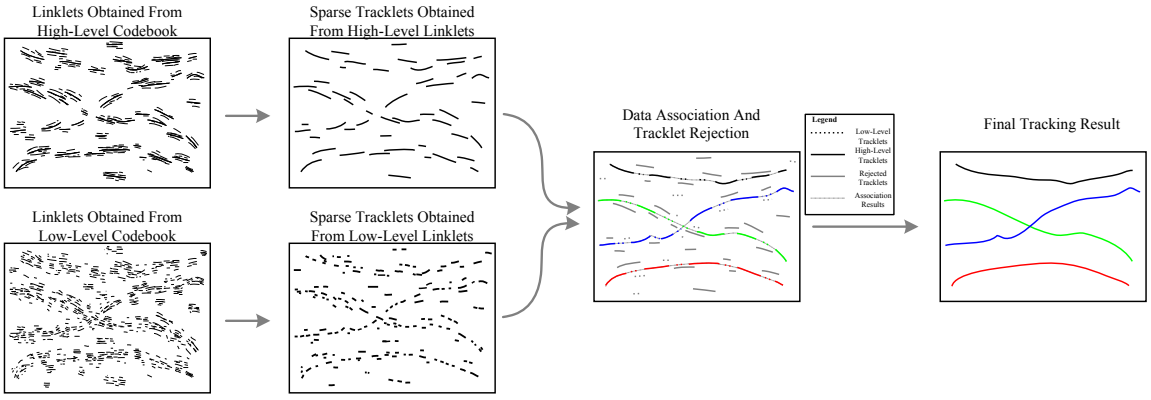


FIGURE 6.1. Overview of the algorithm. The goal is to estimate the trajectory of the moving objects in the video without invoking object detection. Initially two sets of linklets are constructed by chaining; the low-level considers small window fragments, while the high-level analyzes a larger region in order to impose a contextual influence. They are obtained by exploiting an activity understanding system. The resultant tracks (chains) are filtered and replaced by a set of sparse representative tracks, the so-called tracklets. Longer trajectories are then generated by using the Markov Chain Monte Carlo Data Association (MCMCDA) algorithm to solve the Maximum A Posteriori (MAP) problem using tracklet affinities. Thus this procedure uses low-level tracklets to connect high-level tracklets when there is a discontinuity in motion or time.

that only employs hierarchical codebooks to characterize local motion patterns as the *observations*. These hierarchical codebooks are obtained as described by the authors in [91]. In addition, by considering tracklets at two hierarchical levels, the data association algorithm is capable of easily handling missing information. Data association is accomplished by considering temporal continuity and motion consistency of both the low- and high-level tracklets, with the additional option of rejecting irrelevant tracklets.

6.2 Related Work

To date, most of the reported approaches for tracking rely on either robust motion or appearance models of each individual object or on object detection, i.e., they are object-centric. Thus a key assumption is that a reliable object detection algorithm exists [87, 63, 47, 22, 118]. This remains a challenge, particularly in complex

and crowded situations. These methods use the detection response to construct an object trajectory. This is accomplished by using data association based on either the detection responses or a set of short tracks called tracklets that are associated with each detected object [22, 102, 63]. Tracklets are mid-level features that provide more spatial and temporal context than raw sensor data during the process of creating consistent object trajectories. This is then followed by data association stage to link the tracklets into multi-frame trajectories. The issue of associating tracklets across time, the so-called data association, is usually formulated as a MAP problem and has been solved using different methods. For example, network flow graphs and cost-flow networks are employed for data association in [61, 20, 47] to determine globally optimal solutions for an entire sequence of tracklets. Other data association approaches include the Hungarian algorithm [77], maximum weight independent sets [17], the Markov Chain Monte Carlo [81, 10, 102], and the iterative hierarchical tracklet linking methods [22].

On the other hand, there are other tracking algorithms, which are based on local spatio-temporal motion patterns in the scene. More closely related to our approach are those that construct motion models for the moving objects without performing any detection [57, 43, 18, 103, 102]. For example, in [55, 57], Hidden Markov Models are employed to learn local motion patterns that are subsequently used as prior statistics for a particle filter. Alternatively, other methods, such as those in [43] and [3], employ the global motion patterns of a crowd to learn local motion patterns of the neighboring local regions. Individual moving entities are detected by associating similar trajectories based on their features in [18] and [103]. These authors assume that subjects move in distinct directions, and thus disregard possible and very likely local motion inconsistencies between different body parts. Thus a single pedestrian could be detected as a multiple target or multiple individuals as the same target. In order to overcome these difficulties, we analyze trajectories at two hierarchical levels, in which the second level accounts for the inconsistency between local motions of a single object.

Our proposed algorithm provides an alternative to such methods by using local motion patterns and contextual information within a data association framework. In contrast to the aforementioned approaches that attempt to track objects either by detection or learning an appearance model of the objects, our goal is to construct a hierarchical model for all moving objects in a scene. The tracking algorithm described here is based on a MAP data association in which the number of targets and the algorithm parameters are automatically learned. The inputs for the data association framework are two sets of tracklets, the low- and high-level tracklets. The low level tracklets take into account local motion patterns, while those at high-level reflect the contextual information existing in neighboring regions.

6.3 Hierarchical Data Association And Tracking

6.3.1 Observations: Low- And High-Level Codebooks Of Local Motions. Consider the overview in Figure 6.1 and assume that a system capable of producing the linklets (on the left) is available for event description. Our aim is to use the information produced by such a system to detect and track all moving objects in the scene. Here we adopt the hierarchical bag of video words framework developed in [91, 90] for short-term event description. In general, this on-line framework produces two sets of codebooks in real-time and assigns labels to local spatio-temporal video volumes (STVs) based on their similarity, while also considering their spatio-temporal relationships. The hierarchical algorithm dynamically codes a video as both a compact set of individual and ensembles of spatio-temporal volumes. These latter are used to construct a probabilistic model of video volumes and their spatio-temporal compositions (see Figure 6.2).

The first step is to represent a video by a meaningful low-level codebook. Using the framework developed in [91], we determine STVs using dense sampling and then cluster them at each frame based on similarity. We refer to the constructed low-level codebook at this level as $\mathbf{C}^{\mathcal{L}}$, as illustrated in Figure 6.2. The 3D STVs, $v_i \in \mathbb{R}^{n_x \times n_y \times n_t}$ are constructed by assuming a volume of size $n_x \times n_y \times n_t$ around each pixel. Each

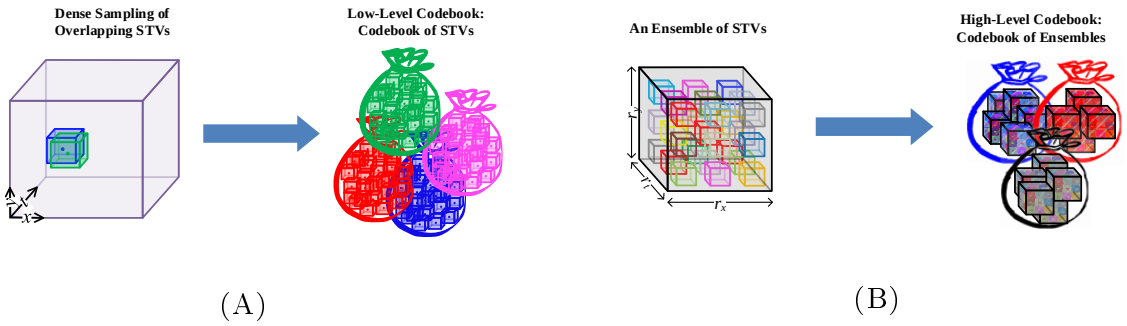
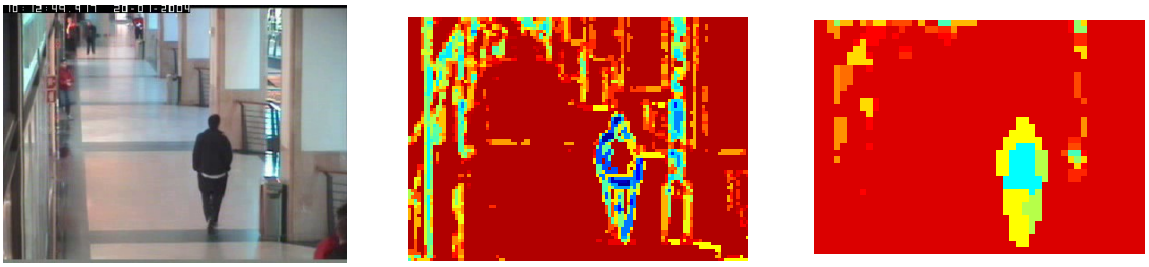


FIGURE 6.2. Observations are represented by low- and high-level codebooks. First, the video is densely sampled scales to produce a set of overlapping STVs and subsequently, a two-level hierarchical codebook is created. (A) At the lower level of the hierarchy, similar video volumes are dynamically grouped to form a conventional fixed-size low-level codebook, $\mathbf{C}^{\mathcal{L}}$. (B) At the higher level, a much larger spatio-temporal 3D volume is created. It contains many STVs at and captures the spatio-temporal arrangement of the volumes, called an ensemble of volumes. Similar ensembles are grouped based on the similarity between arrangements of their video volumes and yet another codebook is formed, $\mathbf{C}^{\mathcal{H}}$ [90, 91].

STV volume is then characterized by a descriptor vector, taken as a histogram of oriented gradients (HOG3D) within the STV. The HOG is constructed using the quantized spatial and temporal gradients converted to polar coordinates and weighted by the gradient magnitude [11, 91, 89]. The codebook is then created using online fuzzy clustering, which is capable of incrementally updating the cluster centers as new data are observed [39]. The clusters are used to produce a codebook of STVs and ultimately assign a label to each STV. Once a video clip has been processed by the first level of clustering as described in the previous section, we examine a large region, R, around each pixel. R contains many video volumes and thereby captures both local and more distant information in the video frames. Such a set is called an *ensemble* of volumes around the particular pixel (Figure 6.2). The *relative* spatio-temporal coordinates of the volume in each ensemble capture the spatio-temporal compositions of the video volumes, [88]. Each ensemble of STVs is represented by a *probability density function* of its spatio-temporal volume distribution, as described in [91]. This histogram becomes the descriptor for each ensemble and forms the



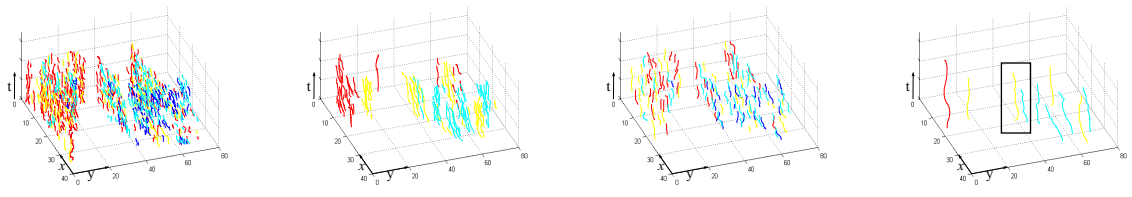
(A) (B) (C)

FIGURE 6.3. Codeword assignment for each pixel. (A) A sample video frame from the CAVIAR dataset [1]; (B) Color-coded low-level codewords assigned to every pixel in the video frame. In this case, there is a large number of low-level codewords; (C) High-level codewords, which represent compositions, are also assigned to every pixel in the video frame. This would generally produce a small number of codewords since it deals with objects in the scene. Each object might be represented by a large number of low-level codewords, while the high-level codebook assigns a few number of codewords to an objects, in most cases one or two.

second level codebook, called the high-level codebook of ensembles of volumes, $\mathbf{C}^{\mathcal{H}}$, as described in [91]. A sample video frame and the assigned codewords are illustrated in Figure 6.3.

6.3.2 Linklets And Tracklets. As mentioned earlier, we use the low- and high-level codebooks to identify the tracklets from the initial observations. In order to do this, we first obtain dense trajectories from these data and then extract a set of sparse trajectories that represents the object motions. We refer to these short-term object motion trajectories as *tracklets*.

Tracklets are obtained from both the low- and high-level codebooks, $\mathbf{C}^{\mathcal{L}}$ and $\mathbf{C}^{\mathcal{H}}$, constructed in 6.3.1. Two codewords are assigned to each pixel $p(x, y)$ at time (t) in the video. Therefore, in a video sequence of temporal length T, a particular pixel $p(x, y)$ is represented by two sequences of assigned codewords called linklets at



(A) (B) (C) (D)

FIGURE 6.4. Linklet and tracklet construction. (A) A set of linklets (short tracks) constructed using observations obtained from the low-level codebook, $\mathcal{X}^{\mathcal{L}}$. (B) A set of linklets constructed using observations obtained from the high-level codebook, $\mathcal{X}^{\mathcal{H}}$. (C) Low-level tracklets, $\mathbf{T}^{\mathcal{L}}$, obtained by grouping similar linklets in $\mathcal{X}^{\mathcal{L}}$. (D) High-level tracklets, $\mathbf{T}^{\mathcal{H}}$, obtained by grouping similar linklets in $\mathcal{X}^{\mathcal{H}}$. The black rectangle indicates the area in XYT-space occupied by a single person. It seems that a single person may produce more than a single trajectory. We expect this because our algorithm does not involve any person or object detection. We deal with this issue in the next section, which describes a data association process that rejects certain tracklets as false positives.

different times[1]:

$$p\left(x,y\right) = \left\{p(x,y) \leftarrow c_i : \forall t \in T,\ c_i \in \mathbf{C}^{\mathcal{L}}\right\}$$
$$p\left(x,y\right) = \left\{p(x,y) \leftarrow c_i : \forall t \in T,\ c_i \in \mathbf{C}^{\mathcal{H}}\right\} \tag{6.1}$$

Given the assigned codewords (labels) for each pixel, we obtain an over-segmented representation of the video (see Figure 6.3). In this over-segmented representation, each segment represents a set of pixels that are similar in terms of local motion patterns. Therefore, it is a simple task to create a short trajectory for each pixel by examining the temporal coherence of its assigned codewords. This is comparable to the concept of so-called "particles" [100]. Here we conservatively associate two responses only if they are in consecutive frames and are close enough in space and similar enough according to their assigned codewords. Thus we obtain, two sets of trajectories, called $\mathcal{X}^{\mathcal{L}}$ and $\mathcal{X}^{\mathcal{H}}$ (see Figure 6.4).

[1] $\leftarrow$ symbolizes value assignment.

Figure 6.4 illustrates the obtained trajectories. It is obvious that the number of linklets is generally more than the number of objects in the scene and that many trajectories might belong to a single object. In addition, we note that the number of linklets created by a single object is much smaller in $\mathcal{X}^{\mathcal{H}}$ than the ones in $\mathcal{X}^{\mathcal{L}}$. Ideally we are interested in obtaining a single trajectory for an object. Thus the linklets belonging to the same object must be merged in order to create a single representative track that describes the motion of the object. Here we follow the idea of clustering trajectories to create a representative object trajectory [100, 19].

Obviously non-informative linklets are removed before constructing clusters of trajectories. These are taken to be relatively motionless or those that carry little information about the motion. They are mainly related to the background or static objects. Similar to [100], we analyze the linklets within a temporal window of the length of T. Then, those trajectories with a small variance are removed[2]:

$$\begin{aligned} \mathbf{X}^{\mathcal{L}} &= \left\{ \mathbf{x} \in \mathcal{X}^{\mathcal{L}},\ var\left\{\mathbf{x}\right\} \geq \varepsilon^{\mathcal{L}} \right\} \\ \mathbf{X}^{\mathcal{H}} &= \left\{ \mathbf{x} \in \mathcal{X}^{\mathcal{H}},\ var\left\{\mathbf{x}\right\} \geq \varepsilon^{\mathcal{H}} \right\} \end{aligned} \tag{6.2}$$

where $\varepsilon^{\mathcal{L}}$ and $\varepsilon^{\mathcal{H}}$ are two thresholds. Clearly, trajectories are not of the same temporal length. Therefore, in order to measure dissimilarity between two trajectories, we adopt the pairwise affinities between all trajectories as introduced in [19]. The distance between two trajectories $\mathbf{x}$ and $\mathbf{y}$, $D\left(\mathbf{x}, \mathbf{y}\right)$, is defined as follows:

$$D^2\left(\mathbf{x}, \mathbf{y}\right) = \max_t \left\{ d_t^2\left(\mathbf{x}, \mathbf{y}\right) \right\} \tag{6.3}$$

where $d_t^2\left(\mathbf{x}, \mathbf{y}\right)$ is distance between two trajectories $\mathbf{x}$ and $\mathbf{y}$ at the time t and defined as follows:

$$d^2\left(\mathbf{x}, \mathbf{y}\right) = \left\|\left(\mathbf{x} - \mathbf{y}\right)\right\|^2 \frac{\left\|\nabla_t\left(\mathbf{x} - \mathbf{y}\right)\right\|^2}{5\sigma_t^2} \tag{6.4}$$

The first factor on the right-hand-side of (6.4) is the average spatial Euclidean distance between the two trajectories. The second factor characterizes the motion of a point

[2] Other methods can be used to remove uninformative codewords, such as the one presented in [89].

aggregated over 5 frames at time t. The normalization term, σ_t, accounts for the local variation in the motion [19]. Given the above distance measurement between two trajectories, clustering is performed using the k-means algorithm. Here we have invoked iterative clustering to determine the optimal number of clusters. In order to perform the merging, we use the Jensen-Shannon divergence measure to compute the actual difference between the resulting clusters. As reported in [100], this method achieves better results than others for trajectory clustering. Clustering produces two sets of low-level tracklets, which we refer to as $\mathbf{T}^{\mathcal{L}}$ and $\mathbf{T}^{\mathcal{H}}$.

As illustrated in Figure 6.4, the tracklets obtained after clustering are not quite reliable for long term object tracking, but do a relatively good job of encoding the moving object motions in the short term. The main advantage of constructing the tracklets based on the two codebooks is that no object detection is required. Although a set of representative trajectories is created for all moving objects in the video, there is no guarantee that an object would be represented by a single trajectory. Moreover, in crowded scenes, the representative trajectories may correspond to more than one object. However, if the motion pattern changes, then the trajectories would separate.

6.3.3 Data Association and High-Level Trajectory Construction. Given the resulting tracklets, high-level trajectories can be generated by linking them in space and time. We achieve this by formulating the data association required as a maximum a posteriori (MAP) problem and solve it with the Markov Chain Monte Carlo Data Association (MCMCDA) algorithm.

The observations are taken to be the constructed tracklets in section 6.3.2:

$$\mathcal{O} = \left\{\mathbf{T}^{\mathcal{L}}, \mathbf{T}^{\mathcal{H}}\right\} \tag{6.5}$$

Let $\mathbf{\Gamma}$ be a tracklet association result, which is a set of trajectories, $\Gamma_k \in \mathbf{\Gamma}$. Γ_k is defined as a set of the connected observations which is a subset of all observations:

$$\Gamma_k = \left\{T_i^{\mathcal{L}}, T_j^{\mathcal{H}}\right\} \subseteq \mathcal{O} \tag{6.6}$$

The goal is to find the most probable set of objects trajectories, $\mathbf{\Gamma}$, which is formulated as a MAP problem:

$$\mathbf{\Gamma}^* = \arg\max_{\mathbf{\Gamma}} P\left(\mathbf{\Gamma}|\mathcal{O}\right) = \arg\max_{\mathbf{\Gamma}} P\left(\mathcal{O}|\mathbf{\Gamma}\right) P\left(\mathbf{\Gamma}\right) \tag{6.7}$$

The likelihood, $P\left(\mathcal{O}|\mathbf{\Gamma}\right)$ indicates how well a set of trajectories matches the observations and the prior, $P\left(\mathbf{\Gamma}\right)$ indicates how correct the data association is. By assuming that the likelihood of the tracklets are conditionally independent, we can rewrite the likelihood, $P\left(\mathcal{O}|\mathbf{\Gamma}\right)$, in (6.7) as follows:

$$P\left(\mathcal{O}|\mathbf{\Gamma}\right) = \prod_{\substack{T_i^{\mathcal{L}} \in \mathbf{T}^{\mathcal{L}} \\ T_j^{\mathcal{H}} \in \mathbf{T}^{\mathcal{H}}}} P\left(T_i^{\mathcal{L}}, T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) \prod_{\Gamma_k \in \mathbf{\Gamma}} P\left(\Gamma_k\right) \tag{6.8}$$

First we consider the encoding of the likelihood of tracklets in (6.8). The observations, that is, the tracklets, can be either true or false trajectories of the object. Therefore, the likelihood of a tracklet, given the set of trajectories, $\mathcal{S}$, can be modeled by a Bernoulli distribution:

$$P\left(T|\mathbf{\Gamma}\right) \sim Bern(p) = \begin{cases} p^{|T|} & : \; T \in \Gamma_k, \; \Gamma_k \in \mathbf{\Gamma} \\ (1-p)^{|T|} & : \; T \notin \Gamma_k, \; \Gamma_k \in \mathbf{\Gamma} \end{cases} \tag{6.9}$$

where $|T|$ denotes how good a tracklet is. Since the tracklets are taken to be clusters of small trajectories constructed in section 6.3.2, $|T|$ is defined as the size of the cluster. Here we assume that the two sets of tracklets, $\mathbf{T}^{\mathcal{L}}$ and $\mathbf{T}^{\mathcal{H}}$, are independent[3]. Therefore, we can write the likelihood in (6.8) as follows:

$$P\left(T_i^{\mathcal{L}}, T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) = P\left(T_i^{\mathcal{L}}|\mathbf{\Gamma}\right) P\left(T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) \tag{6.10}$$

where $P\left(T_i^{\mathcal{L}}|\mathbf{\Gamma}\right) \sim Bern(p^{\mathcal{L}})$ and $P\left(T_j^{\mathcal{H}}|\mathbf{\Gamma}\right) \sim Bern(p^{\mathcal{H}})$ as described in (6.9). This formulation makes it possible to exclude some tracklets from the final data association by assuming that any tracklet can belong to at most one trajectory in the data association process. This is achieved simply by rejecting them as false object tracklets.

[3]The independence assumption is valid here because the consistency between tracklets and observations, i.e., the suitability of the tracklets, is independent of the relationship between trajectories.

Next we consider the encoding of the prior of tracklets in (6.8), $P(\Gamma_k)$. Similar to [22], we model these priors as a Markov chain:

$$P(\Gamma_k) = \prod_{\Gamma_k^t \in \Gamma_k} P\left(\Gamma_k^t | \Gamma_k^{t-1}\right) = P_i\left(\Gamma_k^0\right) P_l\left(\Gamma_k^1 | \Gamma_k^0\right) \ldots P_l\left(\Gamma_k^n | \Gamma_k^{n-1}\right) P_t\left(\Gamma_k^n\right) \tag{6.11}$$

where Γ_k^t is the trajectory of the object at a time instant t. The chain consists of an initialization term, P_i, a probability to link the tracklets, P_l, and a termination probability, P_t, to terminate the trajectory. It is assumed that a trajectory can only be initialized or terminated using the tracklets obtained from the high-level codebook, $\mathbf{T}^{\mathcal{H}}$. Therefore, the probabilities of initializing and terminating a trajectory are written as follows:

$$\begin{aligned} P_i\left(\Gamma_k^0\right) &= P_i\left(T_j^{\mathcal{H}}\right) \\ P_t\left(\Gamma_k^n\right) &= P_i\left(T_j^{\mathcal{H}}\right) \end{aligned} \tag{6.12}$$

The probability of linking two tracklets can be written as:

$$\begin{aligned} P_l\left(\Gamma_k^t | \Gamma_k^{t-1}\right) &= P_l\left(T_{j_t}^{\mathcal{H}}, T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) \\ &= P_l\left(T_{j_t}^{\mathcal{H}} | T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) P_l\left(T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) \end{aligned} \tag{6.13}$$

Two tracklets are linked if they are consistent in the time domain and show similar motion patterns. We assume independence and decompose the probability of linking the tracklets into two probabilities. Therefore (6.13) is rewritten as:

$$\begin{aligned} P_l\left(\Gamma_k^t | \Gamma_k^{t-1}\right) =& P_{\mathcal{T}}\left(T_{j_t}^{\mathcal{H}} | T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) P_{\mathcal{M}}\left(T_{j_t}^{\mathcal{H}} | T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) \\ & P_{\mathcal{T}}\left(T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) P_{\mathcal{M}}\left(T_{i_t}^{\mathcal{L}} | T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) \end{aligned} \tag{6.14}$$

where the temporal consistency probability, $P_{\mathcal{T}}$, is taken to be the hyper-exponential distribution of the temporal gap between the tracklets:

$$P_{\mathcal{T}}\left(T_{j_t}^{\mathcal{H}}|T_{i_t}^{\mathcal{L}}, T_{j_{t-1}}^{\mathcal{H}}, T_{i_{t-1}}^{\mathcal{L}}\right) = \sum_n \alpha_n P_n\left(\tau_n\right)$$

$$P_n\left(\tau_n\right) \sim Exp\left(\lambda_n\right) = \begin{cases} \lambda_n e^{(\lambda_n \tau_n)} & : \tau_n \geq 0 \\ 0 & : \tau_n < 0 \end{cases} \tag{6.15}$$

where τ_n is the temporal distance between the end of a tracklet and the start of its immediate successor. The motion consistency probability, $P_{\mathcal{M}}$, is modeled by assuming that the trajectories follow a constant velocity model and obey a Gaussian distribution.

6.3.4 Markov Chain Monte-Carlo Data Association (MCMCDA) And Parameter Estimation. The combinatorial solution space of $\mathbf{\Gamma}$ in (6.7) is extremely large and finding good tracklet associations is extremely challenging. Here we follow the MCMCDA sampling approach similar to [31, 10] and simultaneously estimate the parameters and $\mathbf{\Gamma}^*$. Figure 6.5 shows how the low- and high-level tracklets can be used for constructing long trajectories in a data association framework.

MCMC is a general method for generating samples from a distribution, p, by constructing a Markov chain in which the states are $\mathbf{\Gamma}$. At any state $\mathbf{\Gamma}$, a new proposal is introduced using the distribution, $q\left(\mathbf{\Gamma}|\mathbf{\Gamma}'\right)$. Following [10], we consider three types of association as a result of the sampling process. The first randomly selects one tracklet and one trajectory. This affects the current state of the tracklet by associating it to the selected trajectory. The second, called swapping, postulates that, all tracklets constructing the two trajectories be swapped at a randomly chosen time. Finally, the third proposes a change of trajectory type. We decide which of the three $\mathbf{\Gamma}'$ should be accepted by employing the Metropolis-Hastings acceptance

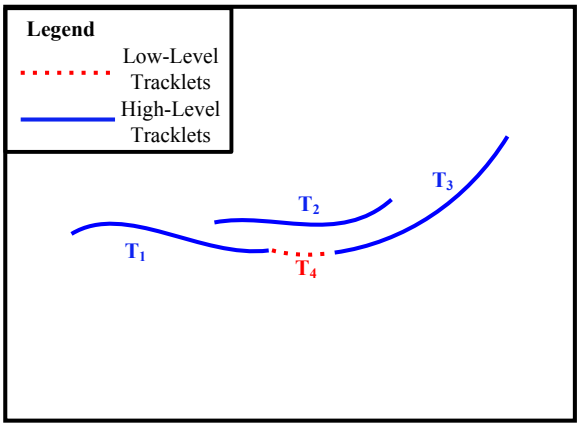


FIGURE 6.5. Data association and tracklet rejection. Formulating the likelihood as described in (6.10) makes it possible to reject some trajectories by considering them as false positives. Here, T_2 is a rejected tracklet. A low-level tracklet, T_4 is used to connect T_1 and T_3 based on motion consistency and temporal continuity.

function [35] which defines the likelihood by:

$$A\left(\mathbf{\Gamma},\mathbf{\Gamma}'\right)=\min\left\{\frac{p\left(\mathbf{\Gamma}'\right)q\left(\mathbf{\Gamma}|\mathbf{\Gamma}'\right)}{p\left(\mathbf{\Gamma}\right)q\left(\mathbf{\Gamma}'|\mathbf{\Gamma}\right)},1\right\} \tag{6.16}$$

In addition, in order to estimate the model parameters described in section 6.3.3, we follow the approach presented in [31]. The latter uses MCMCDA sampling followed by an additional Metropolis-Hastings update for the parameters.

In summary, we have described a method to construct tracklets, given online observations. Then the probability of a tracklet being part of an actual track has been calculated by formulating the data association problem as a MAP estimation. Initial observations are taken to be the low- and high-level codebooks obtained by an event detection system. The low-level codebook codes the local motion patterns, while the high-level codebook codes global motion patterns in videos while considering the scene context. They are then tracked in consecutive frames, which produces two sets of dense tracks of small temporal length, called linklets. These dense linklets are then grouped to produce a small number of representative object tracklets. The representative tracklets are then linked to form long-term object trajectories. The data association framework we have adopted has two main advantages: 1) It can

reject certain tracklets by considering them as parts of false trajectories, and 2) It uses low-level tracklets as supportive information for filling the gaps between high-level tracklets, thereby producing smooth trajectories.

6.4 Experimental Results

The algorithm has been tested using the following publicly available datasets for multiple pedestrian tracking: TUD [4] and CAVIAR datasets [1]. They are taken from static cameras and vary with respect to viewpoint, type of movement, and amount of occlusion. We only use 2D information in all sequences and do not assume any scene knowledge (e.g., ground plane calibration). All parameters have been set experimentally, but most have remained identical for all sequences. In all cases, we have used the suggested parameters in [91] for codebook construction. We show quantitative comparisons with state-of-art methods, as well as visual results of our approach[4].

We follow the same evaluation metrics as those in [118, 102, 61, 128]. These are Mostly Tracked (MT), which is the percentage of the trajectories covered by the tracker output more than 80% of the time; Mostly Lost (ML) which is the percentage of the trajectories covered by the tracker output less than 20% of the time; ID Switch (ID) which is the number of times that a trajectory changes its matched ground truth identity; fragments (FRAG), which is the number of times that a ground truth trajectory is interrupted (i.e., each time it is lost by the current hypothesis); and average False Alarms per Frame (FAF). The results are presented in Table 6.1.

The results indicate that although the correct detections with our algorithm are comparable to the state of the art, they contain more false positives (see Table 6.1). Perhaps one can expect this, since no object detection is employed in our algorithm. Recall that the scene observations that we use are motion descriptors and do not incorporate object appearance, as do object-centric trackers.

[4]See supplemental videos: http://www.cim.mcgill.ca/~javan/index_files/Tracking.html

TABLE 6.1. Comparison of different tracking methods for the CAVIAR [1] and TUD dataset [4].

Dataset	Method	MT	ML	ID	FRAG	FAF
CAVIAR	Hao et al. [23]	84.6	0.7	11	18	0.085
	Yuan et al. [128]	84.6	1.4	11	17	0.157
	Li et al. [61]	85.7	35.7	15	20	-
	Song et al. [102]	84	4	8	6	-
	Ours	84.3	6.4	18	16	0.237
TUD	Yang et al. [118]	70	0	0	1	0.184
	Hao et al. [23]	60	0	0	3	0.014
	Ours	60	10	1	4	0.281

6.5 Summary

In this chapter, we have introduced the use of motion descriptors obtained by an event detection algorithm for multiple object tracking. We have shown how pure motion descriptors for event detection could be employed to build a tracker without requiring an object model. Thus, each individual object is tracked by modeling only the temporal relationships between sequentially occurring local motion patterns. The algorithm is based on the descriptors of moving objects, obtained at two hierarchical levels. By considering both local and global motion patterns, two sets of initial tracks, called linklets, are obtained. Then, a set of sparse tracks, referred to as tracklets, was created by grouping linklets showing similar motion patterns. We then developed associations between them in order to produce longer trajectories.

Although our algorithm possesses no information regarding either an object's color pattern or a human body model, it achieves promising results on challenging data sets. As stated previously, the major drawback of our algorithm is the number of false positives and some problems in maintaining the trajectory identity when objects have similar shape and motion. Further improvements would include incorporating color information to reduce the number of ID switches.

Chapter 7

Closing Remarks

7.1 Summary

In this thesis we started with low level visual features in local spatio-temporal regions as the observations required to build a scene understanding system. This scene understanding system uses motion descriptors to describe and track events involving multiple objects in a scene. In order to achieve these objectives, first, we proposed a new approach for describing contextual information. Scene context is defined as the spatio-temporal relationship between local visual volumes in a large region is space and time, called ensembles of video volumes. To capture the contextual information, the spatio-temporal structures of ensembles are modeled using a hierarchical probabilistic framework. At the lowest level in the data hierarchy, our hierarchical probabilistic structure can be considered as an extension of conventional bag of video words approaches.

Given the hierarchical representation of the scene context, a fully automated abnormality detection algorithm was built. The proposed algorithm uses only the video itself as the training set for the dominant (normal) activities and detects and localizes abnormal regions, while continuously updating the learnt models of visual events. We have shown that this algorithm can effectively localize spatial, temporal, and spatio-temporal abnormal patterns in videos. Experimental results indicates that this method has a competitive performance (in terms of accuracy and computational

cost) compared to the other approaches for anomaly detection. Moreover, it is fast enough for online applications and requires fewer initialization frames. When a separate training set is not available, the algorithm is capable of continuously learning the dominant behavior in an unsupervised manner while simultaneously detecting anomalous patterns. Clearly, this is the preferred behavior for any potential visual surveillance system operating in an unconstrained environment.

As a further analysis, we have shown that the contextual graphs obtained can effectively be used for video-to-video matching and activity recognition. This is achieved by adding another level of processing for obtaining the most informative regions in the scene. Given a single query video (an example of a particular activity), the method computes the similarity of each pixel in each frame of the target videos to the query, and finds the subset of target videos that are most similar to that query. Experimental results are highly competitive with state-of-the-art methods. However, a major advantage of our approach is that it does not require background and foreground segmentation and tracking, and is susceptible to on-line real-time analysis. The proposed video method can easily be extended to multi-action retrieval and action localization by modifying the inference mechanism.

Finally, we have shown how pure motion descriptors for event detection could be employed to build a tracker without even requiring an object model. Thus, each individual object is tracked by modeling only the temporal relationships between sequentially occurring local motion patterns.

7.2 Future Work And Improvements

The proposed algorithms in Chapter 3, 4, and 5 can be considered as similar algorithms with different focuses. One of the major advantages of the proposed algorithm for event recognition in videos is that it does not require a model of the event. However, it does have some drawbacks that need to be addressed in future work. Clearly, such a video representation of activities in a scene is insufficient for directly monitoring long-term and multi-faceted events. This is because, in this thesis,

we made no effort to annotate behaviours that consisted of a number of activities occurring sequentially. Some form of event segmentation would be required to deal with this issue. Future research will extend our approach by adding another level of analysis to the existing hierarchical structure in order to model the spatial and temporal connectivity of the learnt activities.

Although the object tracking method presented in Chapter 6 possesses no information regarding either an object's color pattern or a human body model, it achieves promising results on challenging data sets. As stated previously, the major drawback of our algorithm is the number of false positives and some problems in maintaining the trajectory identity when objects have similar shape and motion. Further improvements would include incorporating color information to reduce the number of ID switches.

In particular, future work can be investigated in several ways as follows:

- Different combinations of feature descriptors for local spatio-temporal video volumes can be employed within the proposed framework. Experimental evaluation of different combinations of feature descriptors is an immediate direction for future work. More specifically, the color information will be considered to improve the tracking algorithm.
- The scene description method presented here does not involve any kind of object detection and recognition. Rather, an assumption has been made about the kind of moving objects in the scene. Therefore, it would be useful to integrate an object recognition method to focus on, for example, only human activities. The current scene description mechanism can be easily extended in this way.
- In order to have an inference mechanism for explaining and annotating more complicated behaviors, object tracking and activity recognition must be performed simultaneously. To achieve this, a particular event could be tracked in the time domain in order to produce a sequence of activities.

These trajectories could then be considered in a generative graphical model to describe the behaviors.

- The current framework can be extended to build a complete scene description algorithm. This could be achieved by creating a sufficiently detailed context of the scene. For example, containing the following information: position of all objects in the scene; the activity of each individual in the scene; the interactions between two individuals, an individual and objects, and a group of people; and the temporal order of the activities. Toward this goal, the first step would be to recognize and analyze all events in a scene, which would be accomplished by using the scene description framework developed in this thesis. The video would then be represented by set of events, localized in space and time, and coded by probabilistic graphical structures. By also including a person detection algorithm, a generative probabilistic graphical structure could be adopted to model the person-event relationship. This would produce a rather concise description of all events and person-event interactions in the scene sufficiently informative for scene analysis and behavior understanding. In order to consider both local and global contextual information, the spatial and temporal relationships between different events and their temporal correspondence would need to be modeled. Although the temporal order of these events can be modeled using Hidden (Semi) Markov Models, the inclusion of spatial relationships would require a more flexible and general structure. A possible solution to this problem is to extend the current generative graphical models (e.g., Probabilistic Latent Semantic Allocation,PLSA) to account for both spatial and temporal context to model the event/event interactions.

APPENDIX A

List of Publications and Patents

A.1 Publications

- M. Javan Roshtkhari and M. D. Levine, "Multiple Object Tracking Using Local Motion Patterns", British Machine Vision Conference (BMVC), 2014.
- M. Javan Roshtkhari and M. D. Levine, "Human Activity Recognition in Videos Using a Single Example", Image and Vision Computing, vol. 31, pp. 864-876, 2013.
- M. Javan Roshtkhari and M. D. Levine, "An On-Line, Real-Time Learning Method For Detecting Anomalies In Videos Using Spatio-Temporal Compositions", Computer Vision and Image Understanding, vol. 117, pp. 1436-1452, 2013.
- M. Javan Roshtkhari, M. D. Levine, "Online Dominant and Anomalous Behavior Detection in Videos", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
- M. Javan Roshtkhari, M. D. Levine, "A Multi-Scale Hierarchical Codebook Method for Human Action Recognition in Videos Using a Single Example", The 9th Conference on Computer and Robot Vision (CRV), 2012.

A.2 Patent

- M. Javan Roshtkhari and M. D. Levine, "Methods And Systems Relating To Activity Analysis", United States US 62/016,133.

Bibliography

[1] Caviar dataset, http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1.

[2] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):555–560, 2008.

[3] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *Computer Vision - ECCV 2008*, volume 5303, pages 1–14. Springer Berlin Heidelberg, 2008.

[4] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1265–1272, 2011.

[5] B. Antic and B. Ommer. Video parsing for abnormality detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2415–2422, 2011.

[6] C. Au, S. Skaff, and J. Clark. Anomaly detection for video surveillance applications. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 888–891, 2006.

[7] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *ICPR*, pages 1–4, 2008.

[8] Y. Benezeth, P.-M. Jodoin, and V. Saligrama. Abnormality detection using low-level co-occurring events. *Pattern Recogn. Lett.*, 32(3):423–431, 2011.

[9] Y. Benezeth, P. M. Jodoin, V. Saligrama, and C. Rosenberger. Abnormal events detection based on spatio-temporal co-occurences. In *Computer Vision and Pattern Recognition(CVPR), 2009 IEEE Conference on*, pages 2458–2465, 2009.

[10] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464, 2011.

[11] M. Bertini, A. Del Bimbo, and L. Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3):320–329, 2012.

[12] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.

[13] A. Bissacco, M. Yang, and S. Soatto. Detecting humans via their pose. *Advances in Neural Information Processing Systems*, 19:169, 2007.

[14] O. Boiman and M. Irani. Detecting irregularities in images and in video. *International Journal of Computer Vision*, 74(1):17–31, 2007.

[15] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1992–1999, 2008.

[16] M. Bregonzio, G. Shaogang, and X. Tao. Recognising action as clouds of space-time interest points. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1948–1955, 2009.

[17] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280, 2011.

[18] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, volume 1, pages 594–601, 2006.

[19] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Computer Vision - ECCV 2010*, volume 6315, pages 282–295.

Springer Berlin Heidelberg, 2010.

[20] A. A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1846–1853, 2013.

[21] B. Chakraborty, M. B. Holte, T. B. Moeslund, and J. Gonzàlez. Selective spatio-temporal interest points. *Computer Vision and Image Understanding*, 116(3):396–410, 2012.

[22] H. Chang, L. Yuan, and R. Nevatia. Multiple target tracking by learning-based hierarchical association of detection responses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):898–910, 2013.

[23] K. Cheng-Hao, H. Chang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 685–692, 2010.

[24] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456, 2011.

[25] H. Dee and S. Velastin. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 19(5):329–343, 2008.

[26] K. G. Derpanis, M. Sizintsev, K. Cannons, and R. P. Wildes. Efficient action spotting based on a spacetime oriented structure representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1990–1997, 2010.

[27] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

[28] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, New York, 2nd edition, 2001.

[29] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733 vol.2, 2003.

[30] E. B. Ermis, V. Saligrama, P. M. Jodoin, and J. Konrad. Motion segmentation and abnormal behavior detection via behavior clustering. In *Image Processing (ICIP), 2008 IEEE International Conference on*, pages 769–772, 2008.

[31] W. Ge and R. T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *British Machine Vision Conference - BMVC*, volume 2, page 5, 2008.

[32] A. Gilbert, J. Illingworth, and R. Bowden. Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In *European Conference on Computer Vision (ECCV)*, pages 222–233. Springer-Verlag, 2008.

[33] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Computer Vision (ICCV), 2009 IEEE International Conference on*, pages 925–931, 2009.

[34] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(99):883 – 897, 2011.

[35] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall, 1998.

[36] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2247–2253, 2007.

[37] N. Haering, P. Venetianer, and A. Lipton. The evolution of video surveillance: an overview. *Machine Vision and Applications*, 19(5):279–290, 2008.

[38] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1933–1940, 2009.

[39] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami. Fuzzy c-means for very large data. *IEEE Trans. Fuzzy Syst.*, PP(99):1–1, 2012.

[40] P. Hore, L. Hall, D. Goldgof, Y. Gu, A. Maudsley, and A. Darkazanli. A scalable framework for segmenting magnetic resonance images. *Journal of Signal Processing Systems*, 54(1):183–203, 2009.

[41] T. Hospedales, S. Gong, and T. Xiang. Video behaviour mining using a dynamic topic model. *International Journal of Computer Vision*, pages 1–21, 2012.

[42] T. M. Hospedales, L. Jian, G. Shaogang, and X. Tao. Identifying rare and subtle behaviors: A weakly supervised joint topic model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2451–2464, 2011.

[43] H. Idrees, N. Warner, and M. Shah. Tracking in dense crowds using prominence and neighborhood motion concurrence. *Image and Vision Computing*, 32(1):14–26, 2014.

[44] M. Javan Roshtkhari and M. Levine. Multiple object tracking using local motion patterns. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[45] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1296–1311, 2003.

[46] Z. Jiang, L. Zhe, and L. S. Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):533–547, 2012.

[47] L. Jingchen, P. Carr, R. T. Collins, and L. Yanxi. Tracking sports players with context-conditioned motion models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1830–1837, 2013.

[48] P. Jodoin, V. Saligrama, and J. Konrad. Behavior subtraction. *IEEE Trans. Image. Proc.*, 21(9):4244–4255, 2012.

[49] P. M. Jodoin, J. Konrad, and V. Saligrama. Modeling background activity for behavior subtraction. In *Distributed Smart Cameras (ICDSC), 2008*

ACM/IEEE International Conference on, pages 1–10, 2008.

[50] Y. Ke, R. Sukthankar, and M. Hebert. Volumetric features for video event detection. *International Journal of Computer Vision*, 88(3):339–362, 2010.

[51] S. Khamis, V. I. Morariu, and L. S. Davis. A flow model for joint action recognition and identity maintenance. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1218–1225, 2012.

[52] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 2921–2928, 2009.

[53] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[54] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2046–2053, 2010.

[55] L. Kratz. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. volume 0, pages 693–700, 2010.

[56] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1446–1453, 2009.

[57] L. Kratz and K. Nishino. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):987–1002, 2012.

[58] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[59] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, volume 1, pages 878–885, 2005.

[60] J. Li, S. Gong, and T. Xiang. Learning behavioural context. *International Journal of Computer Vision*, 97(3):276–304, 2012.

[61] Z. Li, L. Yuan, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008.

[62] J. Liu and M. Shah. Learning human actions via information maximization. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008.

[63] Z. Lu and L. van der Maaten. Structure preserving object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1838–1845, 2013.

[64] V. Mahadevan and N. Vasconcelos. Biologically inspired object tracking using center-surround saliency mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):541–554, 2013.

[65] V. Mahadevan, L. Weixin, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1975–1981, 2010.

[66] M. Marszaek and C. Schmid. Spatial weighting for bag-of-features. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2118–2125, 2006.

[67] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 935–942, 2009.

[68] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 104–111, 2009.

[69] K. Mikolajczyk and H. Uemura. Action recognition with appearance-motion features and fast search trees. *Computer Vision and Image Understanding*, 115(3):426–438, 2011.

[70] A. Mittal, A. Monnet, and N. Paragios. Scene modeling and change detection in dynamic scenes: A subspace approach. *Computer Vision and Image Understanding*, 113(1):63–79, 2009.

[71] B. Morris and M. Trivedi. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 312–319, 2009.

[72] B. T. Morris and M. M. Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2287–2301, 2011.

[73] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14:849–856, 2002.

[74] J. C. Niebles, H. C. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[75] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal localization and categorization of human actions in unsegmented image sequences. *Image Processing, IEEE Transactions on*, 20(4):1126–1140, 2011.

[76] K. Ouivirach, S. Gharti, and M. N. Dailey. Incremental behavior modeling and suspicious activity detection. *Pattern Recognition*, 46(3):671–680, 2013.

[77] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and H. Wensheng. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 666–673, 2006.

[78] M. Piccardi. Background subtraction techniques: a review. In *International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, 2004.

[79] O. P. Popoola and K. Wang. Video-based abnormal human behavior recognition-a review. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, pages 1–14, 2012.

[80] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[81] Y. Qian, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8, 2007.

[82] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. *Advances in Neural Information Processing Systems 16*, 16:1547–1554 1621, 2004. Bbf99 Times Cited:1 Cited References Count:14 Advances in Neural Information Processing Systems.

[83] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[84] K. Rapantzikos, Y. Avrithis, and S. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1454–1461, 2009.

[85] V. Reddy, C. Sanderson, and B. C. Lovell. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Conference on*, pages 55–61, 2011.

[86] E. Ricci, G. Zen, N. Sebe, and S. Messelodi. A prototype learning framework using emd: Application to complex scenes analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, PP(99):1–1, 2012.

[87] A. Roshan Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision - ECCV 2012*, pages 343–356. Springer Berlin Heidelberg, 2012.

[88] M. J. Roshtkhari and M. D. Levine. A multi-scale hierarchical codebook method for human action recognition in videos using a single example. In *Conference*

of Computer and Robot Vision (CRV), pages 182–189, 2012.

[89] M. J. Roshtkhari and M. D. Levine. Human activity recognition in videos using a single example. *Image and Vision Computing*, 31(11):864–876, 2013.

[90] M. J. Roshtkhari and M. D. Levine. An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions, computer vision and image understanding. *Computer Vision and Image Understanding*, 117(10):1436–1452, 2013.

[91] M. J. Roshtkhari and M. D. Levine. Online dominant and anomalous behavior detection in videos. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2609–2616, 2013.

[92] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer Vision (ICCV), 2009 IEEE International Conference on*, pages 1593–1600, 2009.

[93] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1234–1241, 2012.

[94] V. Saligrama and C. Zhu. Video anomaly detection based on local statistical aggregates. In *CVPR*, pages 2112–2119, 2012.

[95] S. Savarese, A. DelPozo, J. C. Niebles, and F.-F. Li. Spatial-temporal correlatons for unsupervised action classification. In *Motion and video Computing (WMVC), 2008 IEEE Workshop on*, pages 1–8, 2008.

[96] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36 Vol.3, 2004.

[97] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *15th international conference on Multimedia*, pages 357–360, Augsburg, Germany, 2007. ACM.

[98] L. Seidenari, M. Bertini, and A. D. Bimbo. Dense spatio-temporal features for non-parametric anomaly detection and localization. In *1st ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pages 27–32. ACM, 2010.

[99] H. Seo and P. Milanfar. Action recognition from one example. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):867–882, 2011.

[100] W. Shandong, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2054–2060, 2010.

[101] E. Shechtman and M. Irani. Space-time behavior-based correlation-or-how to tell if two underlying motion fields are similar without computing them? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(11):2045–2056, 2007.

[102] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *Computer Vision - ECCV 2010*, pages 605–619. Springer-Verlag, 2010.

[103] D. Sugimura, K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1467–1474, 2009.

[104] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 2004–2011, 2009.

[105] T. H. Thi, L. Cheng, J. Zhang, L. Wang, and S. Satoh. Integrating local action elements for action analysis. *Computer Vision and Image Understanding*, 116(3):378–395, 2012.

[106] Y. Tian, L. Cao, Z. Liu, and Z. Zhang. Hierarchical filtered motion for action recognition in crowded videos. *IEEE Trans. Syst., Man, Cybern.*, 42(3):313–323, 2012.

[107] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Ieee Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008. 371XX Times Cited:53 Cited References Count:144.

[108] J. Varadarajan, R. Emonet, and J. Odobez. Bridging the past, present and future: Modeling scene activities from event relationships and global rules. In *CVPR*, pages 2096–2103, 2012.

[109] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[110] H. Wang, A. Klaser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176, 2011.

[111] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference (BMVC)*, 2009.

[112] J. Wang, Z. Chen, and Y. Wu. Action recognition with multiscale spatio-temporal contexts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3185–3192, 2011.

[113] L. Wang and L. Cheng. Elastic sequence correlation for human action analysis. *Image Processing, IEEE Transactions on*, 20(6):1725–1738, 2011.

[114] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[115] A. Wiliem, V. Madasu, W. Boles, and P. Yarlagadda. A suspicious behaviour detection using a context space model for smart surveillance systems. *Computer Vision and Image Understanding*, 116(2):194–209, 2012.

[116] J. Xu, J. Yuan, and Y. Wu. Learning spatio-temporal dependency of local patches for complex motion segmentation. *Computer Vision and Image Understanding*, 115(3):334–351, 2011.

[117] R. Xu and D. Wunsch. *Clustering*. Wiley-IEEE Press, 2009.

[118] B. Yang and R. Nevatia. Multi-target tracking by online learning a crf model of appearance and motion patterns. *International Journal of Computer Vision*, 107(2):203–217, 2014.

[119] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

[120] A. Yao, J. Gall, and L. Van Gool. A hough transform-based voting framework for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2061–2068, 2010.

[121] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[122] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1, Proceedings*, pages 984–989 1223, 2005. Bcr44 Times Cited:5 Cited References Count:13 Proceedings - Ieee Computer Society Conference on Computer Vision and Pattern Recognition.

[123] G. Yu, J. Yuan, and Z. Liu. Unsupervised random forest indexing for fast action search. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 865–872, 2011.

[124] T.-H. Yu, T.-K. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forests. In *Proceedings of the British machine vision conference*, page 56, 2010.

[125] F. Yuan, G.-S. Xia, H. Sahbi, and V. Prinet. Mid-level features and spatio-temporal context for activity recognition. *Pattern Recognition*, 45(12):4182–4191, 2012.

[126] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2442–2449, 2009.

[127] J. Yuan, Z. Liu, and Y. Wu. Discriminative video pattern search for efficient action detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1728–1743, 2011.

[128] L. Yuan, H. Chang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 2953–2960, 2009.

[129] A. Zaharescu and R. Wildes. Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision (ECCV), 2010 European Conference on*, pages 563–576, 2010.

[130] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spatio-temporal phrases for activity recognition computer vision. In *European Conference on Computer Vision (ECCV)*, volume 7574, pages 707–721. Springer Berlin / Heidelberg, 2012.

[131] B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3313–3320, Colorado,Co, 2011.

[132] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2004 IEEE Conference on*, volume 2, pages 819–826, 2004.

[133] X. Zhu and Z. Liu. Human behavior clustering for anomaly detection. *Frontiers of Computer Science in China*, 5(3):279–289, 2011.

[134] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, pages 28–31, 2004.

Mehrsan Javan Roshtkhari

Centre for Intelligent Machines, McGill University, 3480 University St., Montreal (Québec), H3A 2A7, Canada

E-mail address: javan@cim.mcgill.ca